NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

DESIGN, CONSTRUCTION AND ANALYSIS OF A 14-BIT DIRECT DIGITAL ANTENNA UTILIZING OPTICAL SAMPLING AND OPTIMUM SNS ENCODING

by

William P. Ringer

September, 1997

Thesis Advisor: Phillip E. Pace

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
September 1997

3. REPORT TYPE AND DATES COVERED
Master's Thesis

4. TITLE AND SUBTITLE
DESIGN, CONSTRUCTION AND ANALYSIS OF A 14-BIT DIRECT DIGITAL ANTENNA UTILIZINGOPTICAL SAMPLING AND OPTIMUM SNS ENCODING

5. FUNDING NUMBERS

6. AUTHOR(S)
Ringer, William P.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Center for Reconnaissance Research
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Direct digital direction finding (DF) antennas will allow an incoming signal to be digitally encoded at the antenna with high dynamic range (14 bits ≈ 86 dB) without the use of down conversion that is typically necessary. As a shipboard DF device, it also allows for the encoding of wide-band, high-power signals (e.g., ± 43 volts) that can often appear on shipboard antennas due to the presence of in-band transmitters that are located close by. This design utilizes three pulsed-laser driven Mach-Zehnder optical interferometers to sample the RF signal. Each channel requires only 6-bit accuracy (64 comparators) to produce an Optimum Symmetrical Number System (OSNS) residue representation of the input signal. These residues are then sent to a locally programmed Field Programmable Gate Array (FPGA) for decoding into a 14-bit digital representation of the input RF voltage. Modern day FPGA devices are rapidly becoming the state of the art in programmable logic. The inclusion of on-chip flip-flops allows for a fast and efficient pipelined approach to OSNS decoding. This thesis documents the first 14-bit digital antenna which utilizes an FPGA algorithm as a method of OSNS decoding. This design uses FPGA processors for both OSNS decoding and Parity processing.

14. SUBJECT TERMS
Analog-to-Digital Converter, Symmetrical Number System, Field Programmable Gate Arrays (FPGA), Digital antennas, Digital processing for analog-to-digital conversion.

15. NUMBER OF PAGES
134

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT
Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

19. SECURITY CLASSIFI- CATION OF ABSTRACT
Unclassified

20. LIMITATION OF ABSTRACT
UL

Approved for public release; distribution is unlimited

DESIGN, CONSTRUCTION AND ANALYSIS OF A 14-BIT DIRECT DIGITAL ANTENNA UTILIZING OPTICAL SAMPLING AND OPTIMUM SNS ENCODING

William P. Ringer
Lieutenant, United States Navy
BSEE, University of Washington, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September, 1997

ABSTRACT

Direct digital direction finding (DF) antennas will allow an incoming signal to be digitally encoded at the antenna with high dynamic range (14 bits ≈ 86 dB) without the use of down conversion that is typically necessary. As a shipboard DF device, it also allows for the encoding of wide-band, high-power signals (e.g., ± 43 volts) that can often appear on shipboard antennas due to the presence of in-band transmitters that are located close by. This design utilizes three pulsed-laser driven Mach-Zehnder optical interferometers to sample the RF signal. Each channel requires only 6-bit accuracy (64 comparators) to produce an Optimum Symmetrical Number System (OSNS) residue representation of the input signal. These residues are then sent to a locally programmed Field Programmable Gate Array (FPGA) for decoding into a 14-bit digital representation of the input RF voltage. Modern day FPGA devices are rapidly becoming the state of the art in programmable logic. The inclusion of on-chip flip-flops allows for a fast and efficient pipelined approach to OSNS decoding. This thesis documents the first 14-bit digital antenna which utilizes an FPGA algorithm as a method of OSNS decoding. This design uses FPGA processors for both OSNS decoding and Parity processing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDEMENT

This project has evolved over a long time. It stands on the shoulders of giants. This work would not have been possible without the previous work of Lieutenant R. D. Walley, Lieutenant J. L. Shafer, Lieutenant Commander R. Carr and Lieutenant Hiromichi Yamakoshi. Their previous work built a solid foundation for this project. I would like to thank Mr. Henry Chou, PO2 Reynolds and PO1 Bailey. Their assistance in the optics lab was a godsend and their computer expertise is without peer. I would like to thank my lab partner, Lt. Kevin Foster. His presence in the lab was much needed during the times of high stress, which appeared to happen at least once a week. Next I would like to acknowledge Mr. Rick Patterson from Naval Research and Development NRaD who designed the d.c. restoration circuits, d.c. bias circuits and comparator boards for the project. Also System Processes and Engineering Corporation (SPEC) who built the comparator chips.

I would like to personally thank my thesis advisor, Professor Phillip Pace, and my second reader, Professor John Powers. Their guidance and understanding were instrumental in the completion of this thesis.

This work was sponsored inpart by the Space and Naval Warfare Systems Command and the NPS Center for Reconnaissance Research.

Finally I would like to acknowledge my wife, Patricia, and my son, Robert, and my daughter, Rebecca. Their support was the inspiration I needed to complete this project. I love you all.

I. INTRODUCTION

A. REVIEW OF FOLDING ANALOG TO DIGITAL CONVERTERS

The basis for this design of a Direct Digital Antenna (DDA) is the folding analog-to- digital converter. The output from the antenna feed will be a continuously changing time-domain analog signal. This signal would normally be routed to an analog radio frequency (RF) amplifier or filter circuit. These circuits can easily be overdriven by local radio communications equipment emissions. If, instead, the signal from the antenna were converted directly to a digital representation of this analog signal, locally generated interference could be filtered out digitally. Therefore, the need for an analog-to-digital converter (ADC) that directly digitizes the antenna signal arises. There are generally three types of ADCs: the *slow integrating* ADC, good for slowly varying dc voltages; the *successive approximation* ADC, used to digitize audio signals; and the *flash* ADC which is used to digitize video signals. The *flash* ADC has the fastest conversion rate but is also the most costly. [Ref. 1]

The need arises for ADCs with higher resolution and faster conversion speeds. The most popular type of fast converter is the *flash* ADC. This architecture requires 2^N-1 comparators to achieve N-bit resolution. For this project's 14-bit resolution this would be 16,383 comparators. The large number of comparators makes it difficult to align and fabricate. The Optimum Symmetrical Number System (OSNS) can be used to preprocess the analog input signal, thus reducing the number of comparators (for this project only 253 comparators are used).

This thesis presents a prototype design for a DDA system based on an electro-optical folding ADC architecture which incorporates an OSNS encoding scheme [Ref. 2]. The OSNS preprocessing scheme is used to decompose the ADC conversion into a number of parallel sub-conversions (moduli) which are of smaller computational complexity. Each modulus symmetrically folds the analog RF signal with a period equal to twice the modulus. These signals are then processed by separate *flash* ADCs of lesser complexity (for this project only four 6-bit converters are needed). A much higher resolution is achieved after the *N* different OSNS moduli are used and the results of the lower precision sub-conversions are recombined. The parallel use of folding circuits increases the ADC resolution without increasing the folding rate of the system.

B. DIRECT DIGITAL ANTENNA OVERVIEW

A direct digital antenna provides a means of digitizing the RF analog signal as it appears on the output of the antenna feed. A block diagram of a shipboard DDA design is shown in Figure 1.

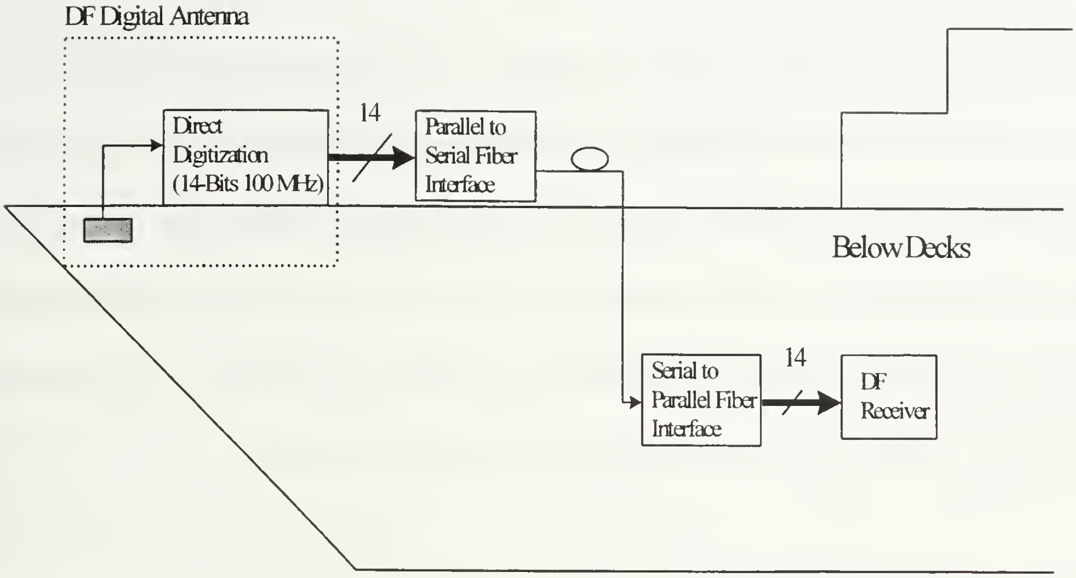

DF Digital Antenna
Direct Digitization (14-Bits 100 MHz)
14
Parallel to Serial Fiber Interface
Below Decks
Serial to Parallel Fiber Interface
14
DF Receiver

Figure 1 Direct Digital Antenna Concept for Shipboard DF.

The RF from the antenna feed is fed directly to the direct digitization circuit comprised of a wideband electro-optic folding ADC which converts the antenna signal to a 14-bit digital representation of the input. The 14-bit signal is then fed to a parallel-to-serial fiber interface, sent below to a 14-bit serial-to-parallel interface and fed to the digital direction finding (DF) receiver for final processing. The digital receiver can also be fed digital inputs representing local interference signals which can then be filtered out of the input from the DDA.

A block diagram of a three-channel wideband SNS digital antenna is shown in Figure 2. For each channel, the received RF energy (picked up by the antenna) is passed through a Low Pass Filter (LPF) for anti-aliasing and then fed through a DC bias/attenuation circuit and applied to a reflective parallel configuration of three Mach-Zehnder interferometers (MZIs). The optical input to the three interferometers is provided by a pulsed laser (6 ns pulse at a pulse repetition interval of 200 ns) that samples the RF signal. The optical pulse is sent through a 1 x 4 optical power splitter, through three circulators and on to the respective interferometers. The RF input signal applied to the interferometers modulates the amplitude of the laser pulse and is scaled by a function of the interferometer's V_π. The modulated laser pulse is then detected, applied to a DC restoration amplifier circuit and then amplitude analyzed by a bank of m-1 comparators where m (m = 63, 64 and 65 in this system) is the channel modulus.

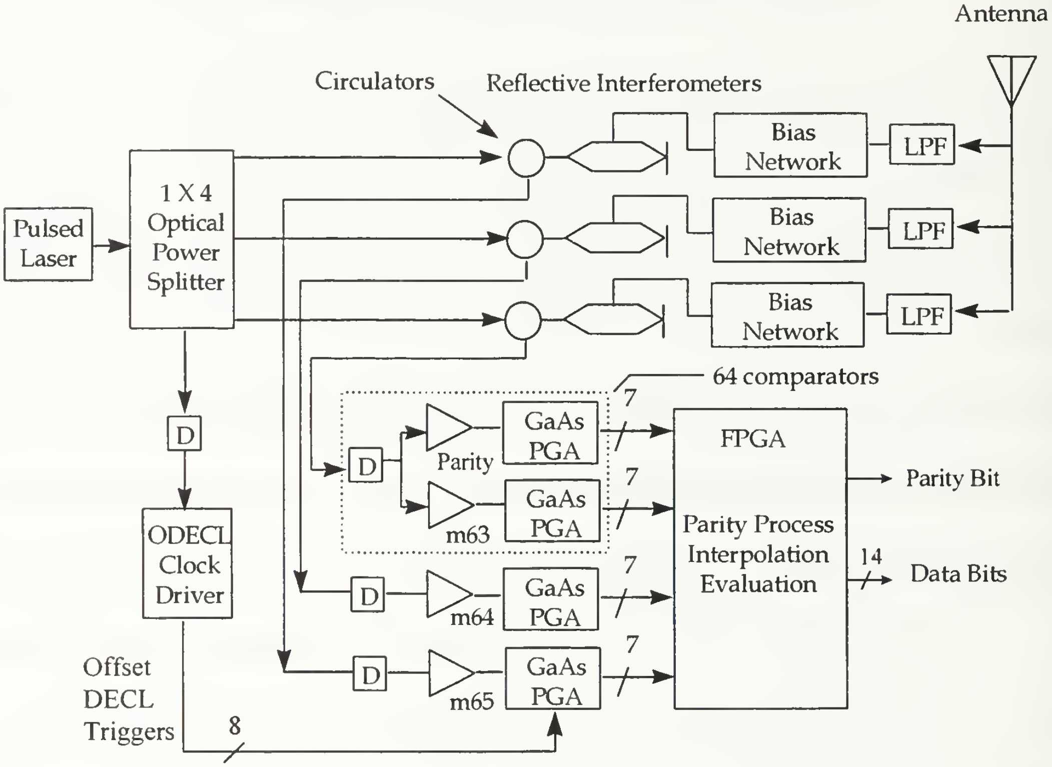


Figure 2 14-bit Direct Digitizing circuit.

The binary representation of the resulting comparator thermometer code for each channel is sent to a Field Programmable Gate Array (FPGA) circuit for conversion to a 14-bit digital representation of the input analog signal. The timing of the digital encoding is controlled by an Offset Differential Emitter Coupled Logic (ODECL) adjustable trigger delay circuit. Note that the m=63 channel requires an additional m=63 comparators that assist in isolating possible encoding errors that might occur. These errors can then be corrected in the signal processing

The optical front-end of this circuit is discussed in Reference 3 and the ODECL circuitry is discussed in Reference 4. This thesis will concentrate on the design of the comparator circuitry and the FPGA devices.

C. PRINCIPAL CONTRIBUTIONS

This thesis fully develops a decoder designed to translate the Optimum SNS (OSNS) residues into an equivalent digital representation of the input shipboard DF antenna voltage.

This decoder was designed and implemented using a Field Programmable Gate Array (FPGA) device and employs a unique memoryless OSNS-to-binary conversion process discussed in Reference 5. This design uniquely applies an optical processor utilizing three Mach-Zehnder interferometers as a convenient preprocessing interface to the receiving antenna. This thesis focuses primarily on the design and construction of the comparator circuitry and the design of the FPGA processing algorithm.

The comparator boards were programmed using the National Instruments LabVIEW instrumentation programming language and the FPGA design was created using ALDEC's Foundation ActiveCAd program and Xilinx's XactStep FPGA programming software.

D. THESIS OUTLINE

Chapter II of this thesis consists of an overview of the optical portion of the direct digitizing circuit and introduces the reader to some of the formulas that comprise the basis for the design. Chapter III discusses the design and construction of the comparator boards including a description of the LabVIEW software used to program them. Chapter IV describes the algorithm used to implement the OSNS-to-binary conversion and includes an in-depth discussion of the design created using Foundation ActiveCAD. Chapter V explores the timing requirements necessary for system synchronization. Chapter VI presents the results of the research including outputs from the optical subsection, the dc-restoration circuitry, the ODECL trigger circuitry, the comparator boards and the final output from the FPGA circuits. Finally, Chapter VII demonstrates the performance of the total system, sums up the findings, and forms conclusions based on the data.

II. OPTICAL SUBSECTION

A. OVERVIEW

A block diagram of the optical subsection of the direct digitizing circuit is shown in Figure 3.

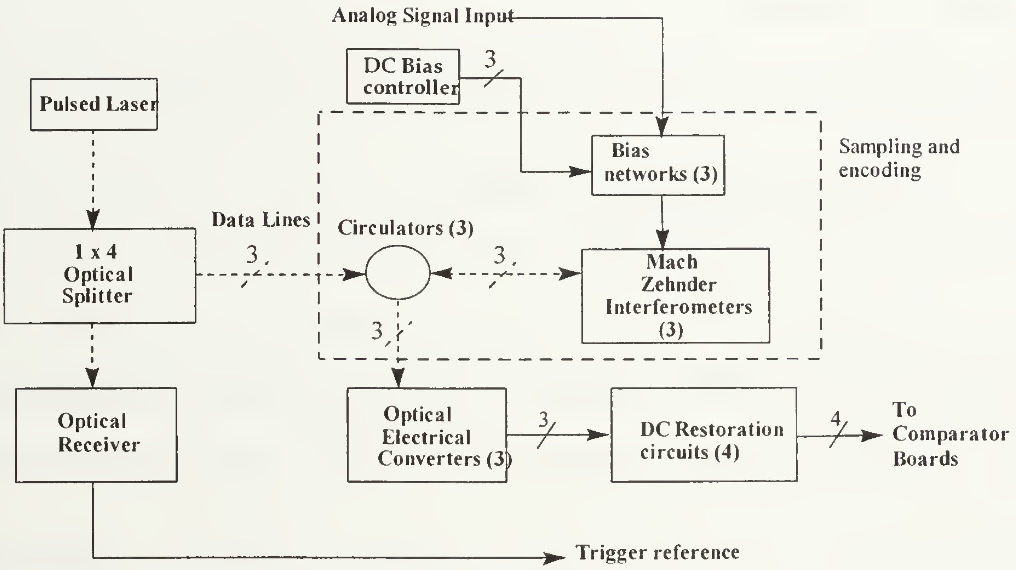


Figure 3 Optical Subsection.

Precise and efficient sampling is accomplished using a high speed semiconductor laser source which is triggered by a Hewlett-Packard HP-71600B bit-error encoder. The laser pulses are split off into four parallel signals, three of which are used to sample the RF signal while the fourth is used as a reference pulse for the ODECL adjustable trigger delay circuit that latches the output the comparator array. The analog RF signal to be sampled is fed from the antenna into the three bias-termination networks. These distribute the input signal to the three optical interferometers of the digital antenna and have adjustments for setting the moduli of the OSNS and for phase shifting the interferometer transfer functions in order to align the folding waveforms across the three channels. The phase (*$\Delta\phi$*) of the MZI folded waveform is a function of the DC voltage level on the electrode. Thus, DC bias on the electrodes provide a means to align the minimums of the parallel outputs in accordance with the OSNS [Ref. 6]. The normalized output of the MZI is given by:

$$\mathrm{I}\,(v, m_i) = \tfrac{1}{2} + \tfrac{1}{2}\cos\left(\frac{\pi}{m_i}v + \Delta\phi_i\right) \qquad (1)$$

where m_i is the desired modulus number, v_i is the desired level within the modulus (from 1 to m_i–1), and the d.c.-voltage-dependent phase shift $\Delta\phi_i$ is used to phase the transfer function within each channel correctly. Three bias networks were also incorporated into the circuitry in order to provide a means of adjusting the MZI folding periods in accordance with the desired moduli ratios. Pulse amplitude modulation and OSNS folding is performed by the three parallel, reflective MZIs. The laser sampling pulse is routed into the MZI via a circulator and modulated by the incoming RF signal in accordance with the linear Pockel's effect. The pulsed signal is encoded into the OSNS and reflected back through the circulator and onto a detector/amplifier circuit.

B. OSNS BASICS

The Optimum Symmetrical Number System (OSNS) scheme is composed of a number of pairwise relatively prime (PRP) moduli m_i. The integers within each SNS modulus are representative of a symmetrically folded waveform with the period of the waveform equal to twice the PRP modulus, i.e., $2m_i$. For a given m, the integer values within the folded waveform (symmetrical residues) are given by the row vector

$$\bar{x}_m = \left[0,1,\cdots,m-1,m-1,\cdots,1,0\right]. \qquad (2)$$

Figure 4 shows the optimum SNS folding waveforms and SNS output codes for both $m_1 = 4$ and $m_2 = 5$. The x axis represents the normalized input voltage. The "T" values along the left side of the figure represent predetermined voltage levels which are used as reference levels for a bank of comparator circuits. The numbers in squares at the top of the figure represents the number of comparators turned on for the given input voltage band. (These values are also known as residues.) From Eq. (2) we can observe that the required number of comparators for each channel is m_i - 1. Due to the presence of ambiguities, the integers within Eq. (2) do not form a complete system of length $2m$ by themselves. For example, for $i = 5$, the first and last residue (for inputs between 0 and 1 and 9 and 10, respectively) will each be 0 which does not allow for unambiguous backward translation of the residues to recover the input. By recombining N channels, however, the OSNS is rendered a complete system having a one-to-one correspondence with Residue Number System (RNS). [Ref. 5] For N equal to the number of PRP moduli, the dynamic range M of the system is given by

$$M = \prod_{i=1}^{N} m_i \ . \tag{3}$$

This dynamic range is also the position of the first repetitive moduli vector. For example, for $m_1 = 4$ and $m_2 = 5$, the first repetitive moduli vector occurs at an input of 20 as shown in Table 1.

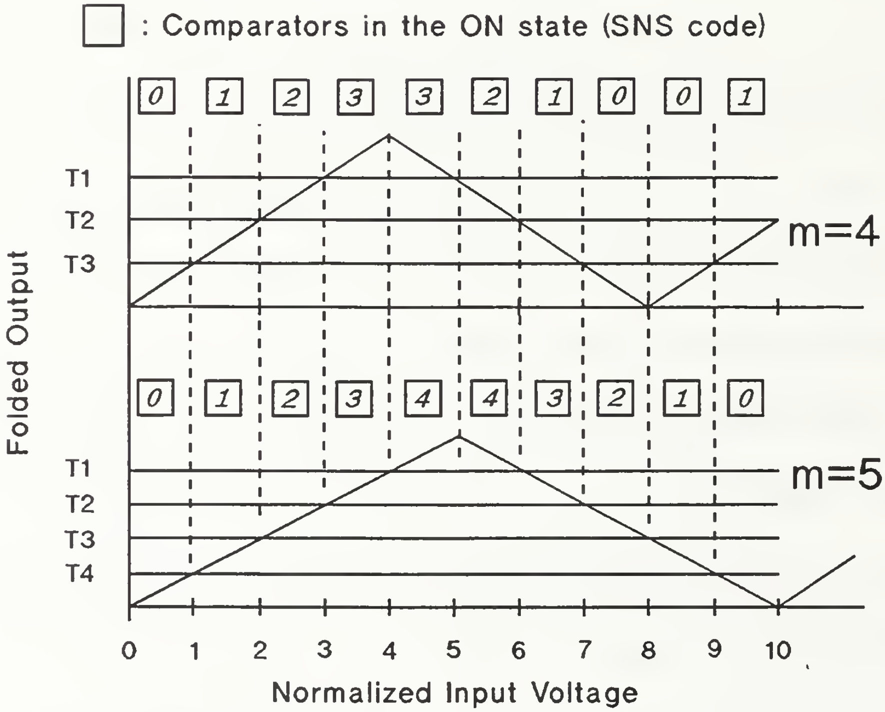


Figure 4 OSNS folding waveforms and output codes for $m_1 = 4$ and $m_2 = 5$.

In an OSNS analog preprocessor, a number system, based on N different moduli, is incorporated that will produce the desired dynamic range from (3). An input signal is applied to the N different moduli in parallel. Each modulus corresponds to a folding circuit that folds the input signal with a period equal to twice the value of the modulus. The folded waveform that is produced as the output of each folding circuit is then quantized by the comparator boards. The output of the comparator boards is a binary number representing the signal in an OSNS format.

Normalized Input (q = 1)	m = 4	m = 5
0	0	0
1	1	1
2	2	2
3	3	3
4	3	4
5	2	4
6	1	3
7	0	2
8	0	1
9	1	0
10	2	0
11	3	1
12	3	2
13	2	3
14	1	4
15	0	4
16	0	3
17	1	2
18	2	1
19	3	0
20	3	0

Table 1. OSNS vectors for $m_1 = 4$ and $m_2 = 5$.

The goal of this thesis was to create a 14-bit OSNS ADC. In order to provide 14-bit resolution, a dynamic range (M) of 2^{14} or 16,382 is necessary. The moduli chosen to provide this dynamic range are $m_1 = 63$, $m_2 = 64$ and $m_3 = 65$. Using Eq. (3) to calculate the dynamic range provided by the OSNS encoding yields a value of 262,080, which is well above the required 16,382. The minimum modulus (m_1=63) is chosen by considering the number of folds available from the optical interferometers. A derivation of the minimum modulus calculation is contained in Reference 2.

C. PARITY

In order for the Optimum SNS system to decode the input magnitude of the signal properly, the comparators for the different moduli (channels) have to switch at the same instant. This is shown in Figure 4 which depicts the comparator encoding scheme for a simple two moduli (m_1= 4, m_2 = 5) encoder. If the comparators do not switch exactly at each LSB transition, a possible encoding error will occur.

A fourth (parity) comparator circuit (m_1+1 comparators) has been established to provide an indication of whether or not the sample has occurred within the region where a slight offset of the comparator alignment may occur. This is accomplished by the creation of isolation bands. Using the smallest modulus m_1-1 comparators and the m_1+1 parity comparators, parity bands can be established around the code transition points. If a sample occurs within one of the isolation bands, the signal may be unusable and interpolation processing will be performed on that sample. The parity encoding scheme for the simple modulus 4 and 5 encoder is shown in Figure 5.

Determination of whether the sample has occurred within a parity band is made using a simple EXCLUSIVE-OR logic comparison to verify the number of comparators in the mod-63 and parity channels that are turned on. An even number of comparators (even parity) indicates the sample is possibly bad. An odd number of comparators (odd parity) indicates the sample is good.

This chapter provided a background for OSNS systems. The next chapter will discuss the comparator boards and the programming required to ensure the proper comparator levels for successful decoding of the OSNS residues are routed to the correct comparators.

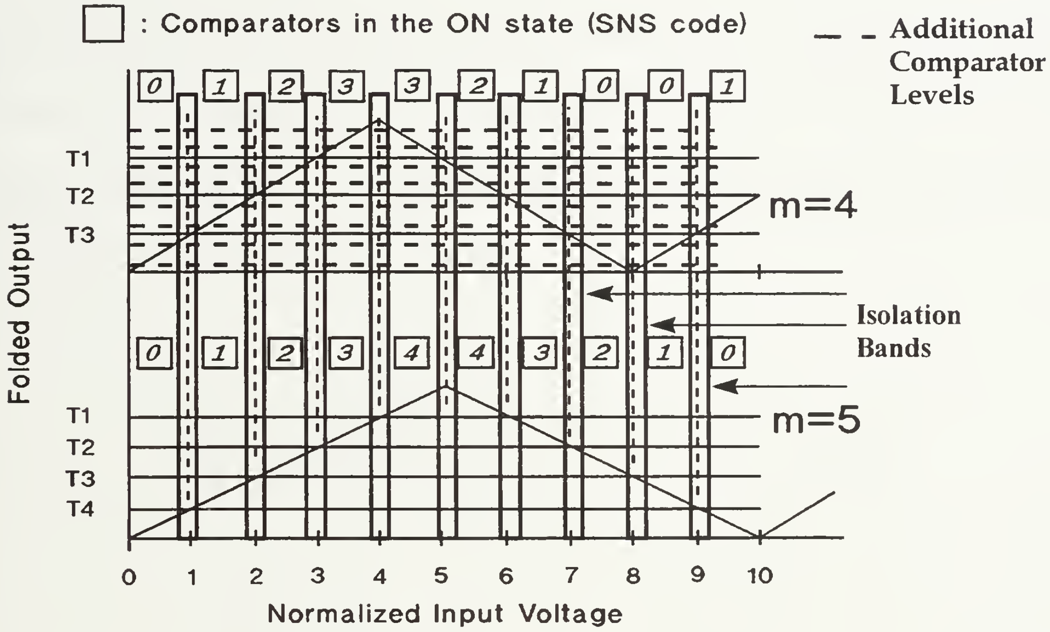


Figure 5 Isolation bands.

III. COMPARATOR BOARDS

The comparator boards are used to quantize the analog voltage pulses coming from the dc restoration circuit. There are four comparator boards, one each for the mod 63, 64 and 65 outputs as well as one for parity processing. The two major subsections of the boards are the reference voltage generation subsection and the comparator subsection with binary output encoding. A block diagram of a comparator board is shown in Figure 6. Discussions of these subsections follows.

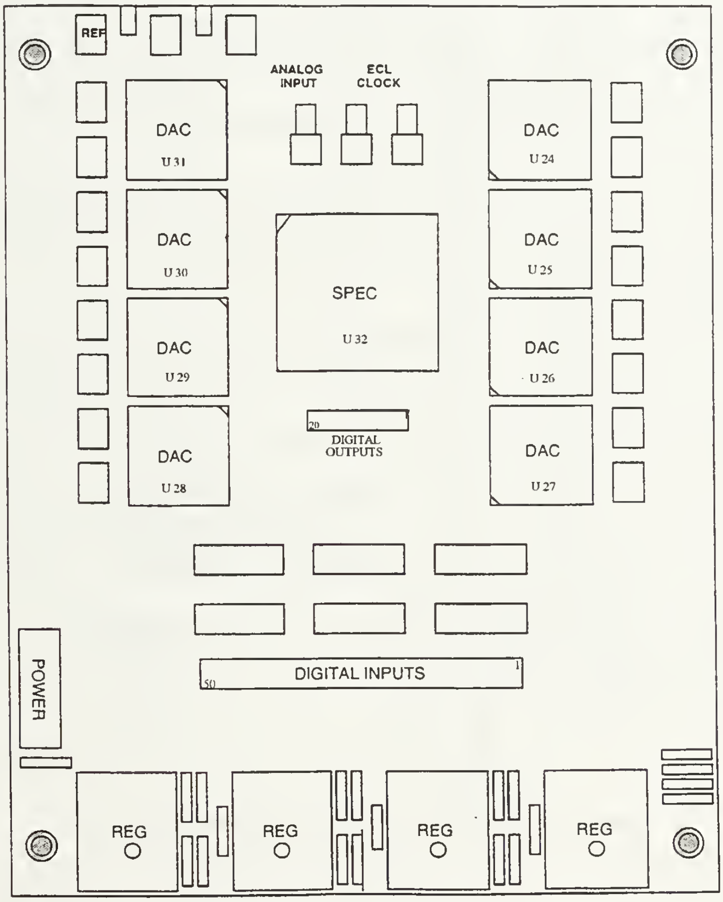


Figure 6 Comparator board block diagram.

A. REFERENCE VOLTAGE GENERATION

Reference voltage generation is accomplished using the MAX547 Octal 13-bit Digital-to-Analog Converters (DAC) made by Maxim Integrated Products. Eight of these DACs are installed on each comparator board. A block diagram of a MAX547 is shown in Figure 7. The usage of the MAX547 to produce precision references alleviates the need for resistor ladders and the associated calibration requirements for the resistors. [Ref. 2]

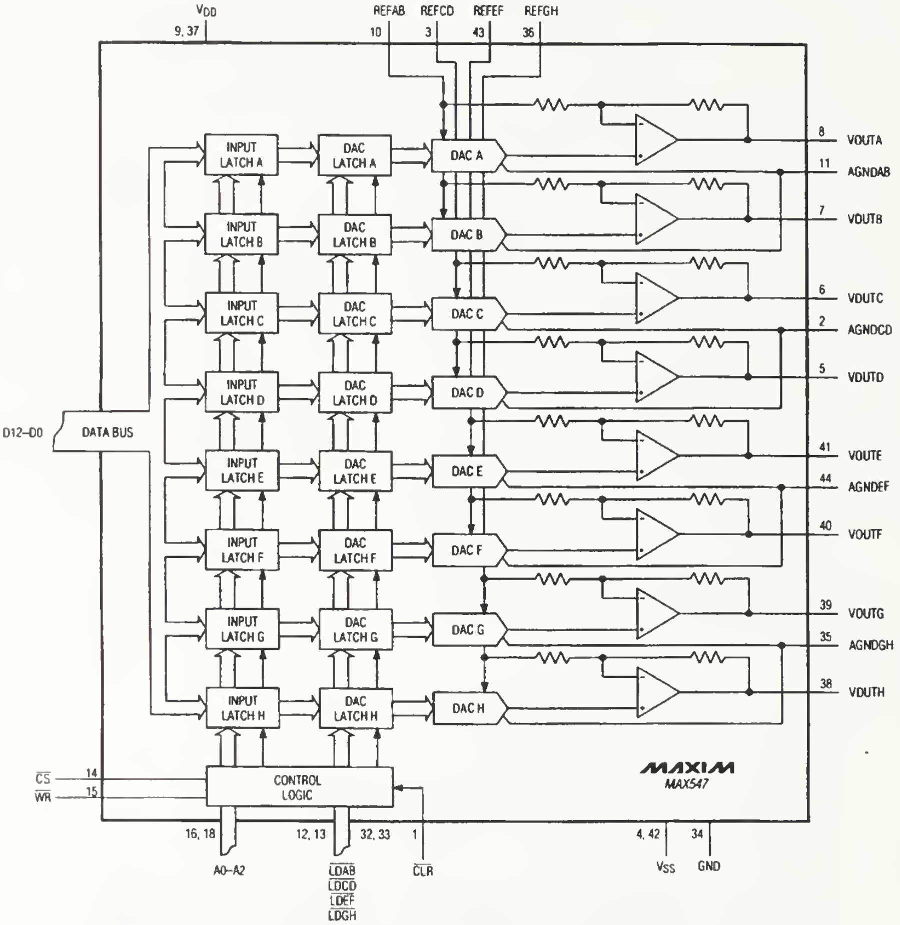
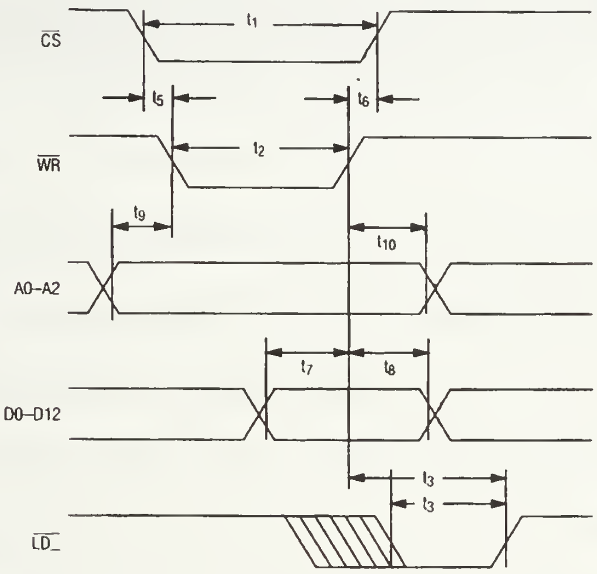


Figure 7 MAX547 block diagram. From Ref. [9].

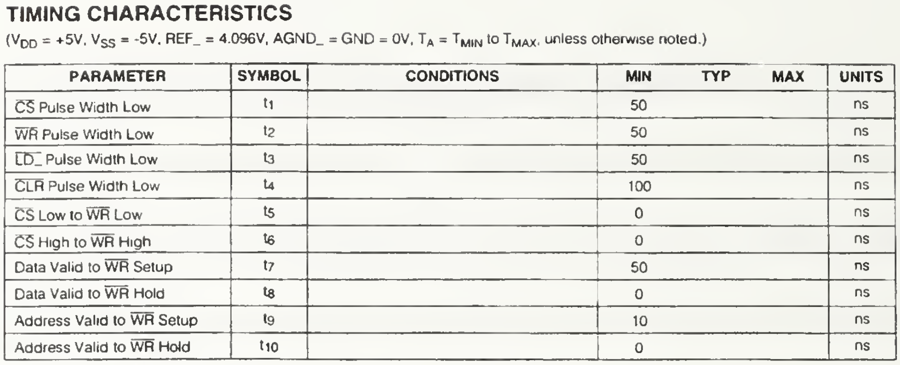
Each DAC is capable of providing eight reference voltages simultaneously (VOUT A-H). This allows 64 reference voltages per board. The mod-65 board uses 64 reference voltages, the mod-64 board uses 63, the mod-63 board uses 62, and the parity board uses 64. The timing diagram required for programming the MAX547 is shown in Figure 8 and a description of the timing constraints is contained in Table 2. Note that t_5, t_6, t_8 and t_{10} can all be 0 ns.



NOTES:

1. ALL INPUT RISE AND FALL TIMES MEASURED FROM 10% TO 90% OF +5V. $t_r = t_f = 5$ns.
2. MEASUREMENT REFERENCE LEVEL IS ($V_{INH} + V_{INL}$)/2.
3. IF $\overline{LD\_}$ IS ACTIVATED WHILE $\overline{WR}$ IS LOW THEN $\overline{LD\_}$ MUST STAY LOW FOR t_3 OR LONGER AFTER $\overline{WR}$ GOES HIGH.

Figure 8 MAX547 write-timing cycle. From Ref. [9].



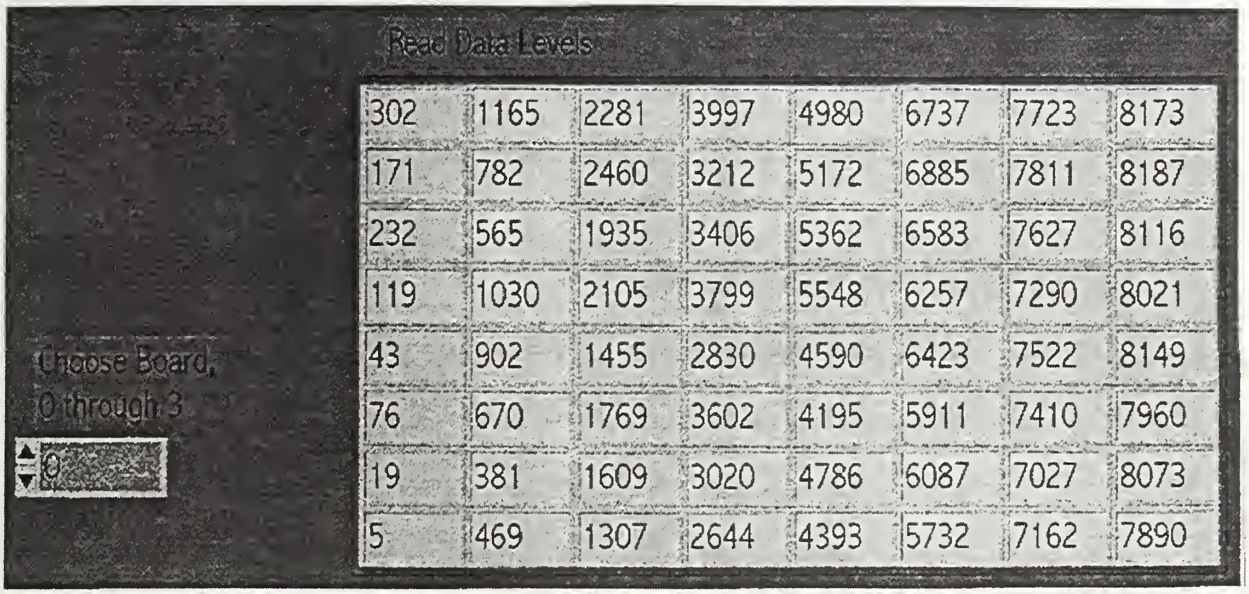
TIMING CHARACTERISTICS

(V_{DD} = +5V, V_{SS} = -5V, REF_ = 4.096V, AGND_ = GND = 0V, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
$\overline{CS}$ Pulse Width Low	t_1		50			ns
$\overline{WR}$ Pulse Width Low	t_2		50			ns
$\overline{LD\_}$ Pulse Width Low	t_3		50			ns
$\overline{CLR}$ Pulse Width Low	t_4		100			ns
$\overline{CS}$ Low to $\overline{WR}$ Low	t_5		0			ns
$\overline{CS}$ High to $\overline{WR}$ High	t_6		0			ns
Data Valid to $\overline{WR}$ Setup	t_7		50			ns
Data Valid to $\overline{WR}$ Hold	t_8		0			ns
Address Valid to $\overline{WR}$ Setup	t_9		10			ns
Address Valid to $\overline{WR}$ Hold	t_{10}		0			ns

Table 2 MAX547 timing characteristics. From Ref. [9].

For programming the matching reference voltages generated by the DACs, the LabVIEW v4.0 graphical instrumentation software package was used. LabVIEW allows the user to create a virtual instrument (vi) which serves as an interface between a computer (for this thesis a Pentium-166 MHz IBM compatible) and equipment external to the computer. Also needed was a DIO-96 interface board which allowed an interconnection between the comparator boards and the computer. Figure 9 shows the front panel for the *SNS_run* vi used to program the DACs. This vi allows the operator to select the comparator board to be programmed where board 0 = mod-65, board 1 = mod-64, board 2 = mod-63 and board 3 = parity. The values are then read from a file stored on the computer hard drive and sent to the comparator board and also displayed on the front panel of the vi as shown in the data levels matrix. Appendix A contains the MATLAB program that was used to calculate the comparator levels and descriptions of the LabVIEW vis used to program the comparator boards.



Read Data Levels

302	1165	2281	3997	4980	6737	7723	8173
171	782	2460	3212	5172	6885	7811	8187
232	565	1935	3406	5362	6583	7627	8116
119	1030	2105	3799	5548	6257	7290	8021
43	902	1455	2830	4590	6423	7522	8149
76	670	1769	3602	4195	5911	7410	7960
19	381	1609	3020	4786	6087	7027	8073
5	469	1307	2644	4393	5732	7162	7890

Choose Board, 0 through 3

0

Figure 9 *SNS_run* **LabVIEW vi front panel.**

The programming sequence followed by the *SNS_run* vi is as follows:

1. Read the desired comparator values from the hard drive.

2. Configure the DIO-96 interface board.

3. Set the following initial values:

   Chip = 0

   DAC = 0

   Board Disable = 1 (must be 0 to enable board)

   Data Level = 0

   Load Enable = 1 (must be 0 to enable loads)

   Write = 1 (must be 0 to enable writes)

   Task ID = Task ID from DIO-96 configuration

4. Select DAC and data values:

   Board Disable = 0 (enables the board).

Chip = correct chip number.

DAC = correct DAC number.

Data Level = correct data level.

5. Set Write = 0 to enable write to input latch.
6. Set Write = 1 to isolate input latch.
7. Set Load = 0 to write to DAC latch.
8. Set Load = 1 to isolate DAC latch.
9. Reset values to those in step 1.
10. Increment DAC number and repeat steps 3 - 9 until DAC = 8.
11. Increment Chip number and repeat steps 3 - 10 until all 64 comparator levels set.

The voltage level generated by the individual DAC is given by the following [Ref. 9]:

$$VOUT = AGND\_ + \left(\frac{D}{4096} - 1\right)\left(V_{DR}\right),$$

where

$AGND\_ = 2$ volts,

D = Data input from LabVIEW program,

$V_{DR} = REF\_ - AGND\_$ (full swing voltage of comparator), and

$REF\_ = 3$ volts.

The comparator data inputs and the corresponding reference voltage levels for each of the DACs is listed in Chapter VI. It is important to note that even though the discussion of the OSNS waveforms in Chapter II showed a continuous moduli waveform, the actual output from the interferometers corresponds to a sampled version of Eq. (1).

The reference voltage values have to be chosen to convert the sampled sinusoidal waveform from the interferometers to OSNS symmetrical residues waveform prior to processing by the FPGA. The MATLAB programs to compute the reference voltage levels are listed in Appendix A.

B. COMPARATOR SUBSECTION AND OUTPUT ENCODING

The 64 voltages generated by the DACs are routed to the SPEC chip (U32) to be used as reference levels for the 64 comparator circuits. Internally, the SPEC chip compares the pulsed analog input voltage from the dc restoration circuit to the reference levels and produces a thermometer coded output. This thermometer code is then translated to a binary representation of the number of comparators whose reference voltages are below or equal to the magnitude of the analog input voltage. This binary representation is then latched at the output of the SPEC chip by the ODECL trigger circuit discussed earlier. Figure 10 shows a block diagram of the comparator subsection and output encoding section. A pin-out of the SPEC chip showing the inputs and outputs is shown in Figure 11.

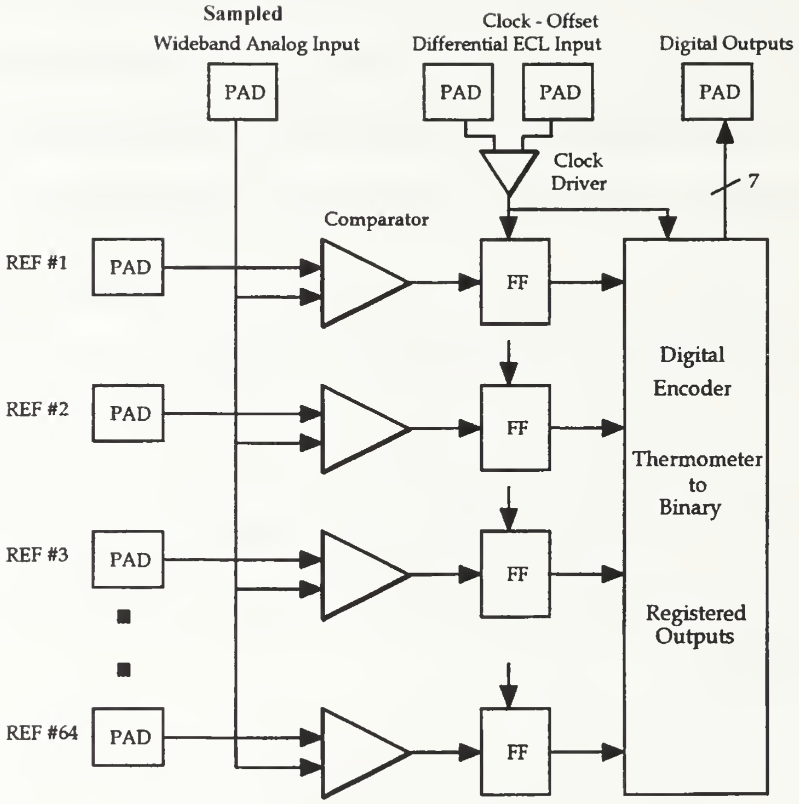


Figure 10 SPEC chip comparator and output encoder block diagram.

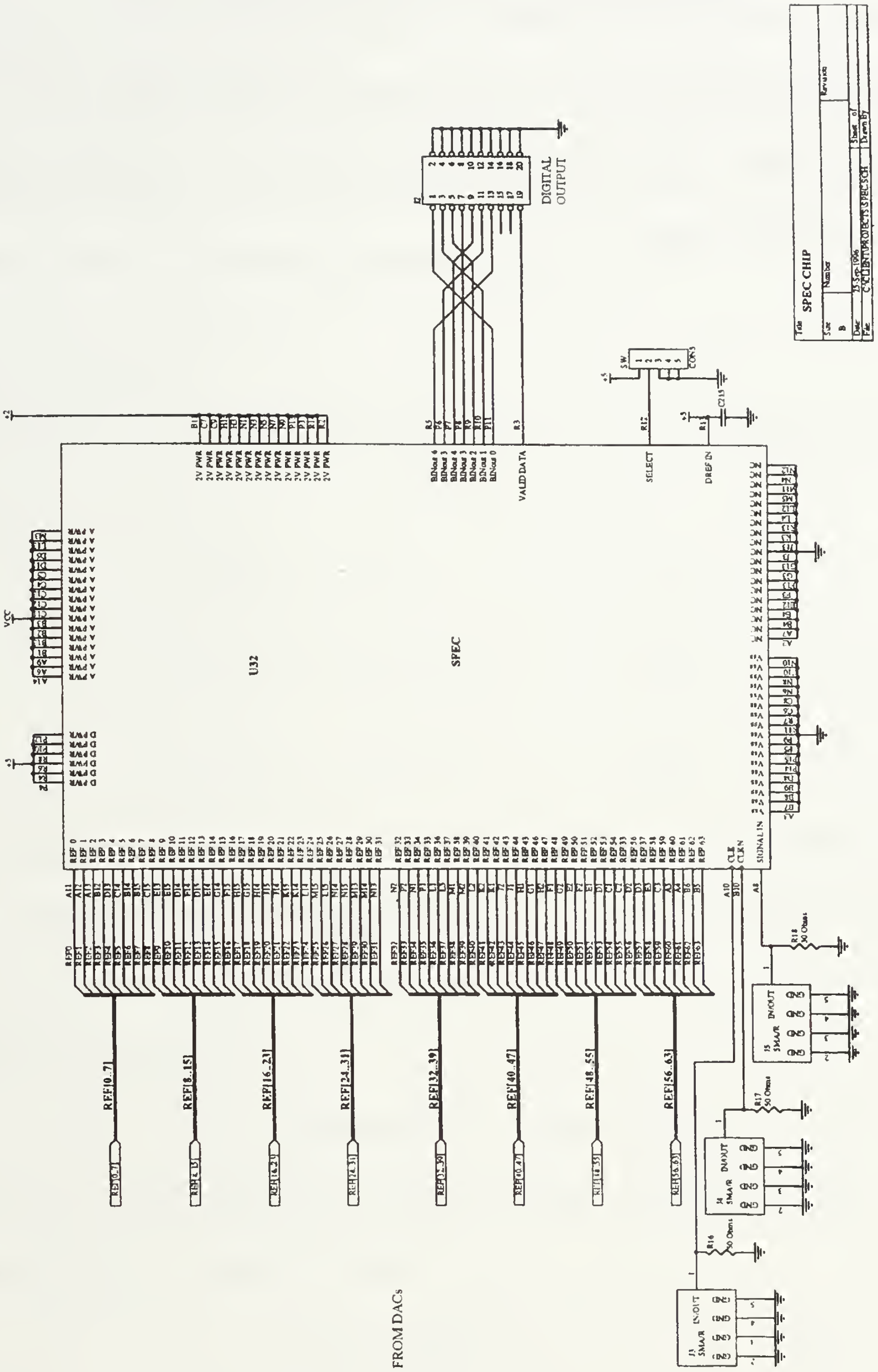


Figure 11 SPEC chip pin-out.

The output of the comparator boards consists of two sets of 6-bit binary digital outputs, for the mod-63 and mod-64, and two sets of 7-bit binary digital outputs for the mod-65 and parity. These digital outputs are routed to the OSNS FPGA which is discussed in the next chapter.

IV. OSNS TO BINARY CONVERSION

This chapter will develop hand-in-hand both the algorithm for converting OSNS symmetrical residues to a binary output and the FPGA schematic design to realize the algorithm. Reference 5 gives further information on the algorithm.

A. OVERVIEW

Consider an OSNS system with moduli $m_1 = 2^k + 1$, $m_2 = 2^k$ and $m_3 = 2^k - 1$ (for this project $k = 6$). Let

$$s_1 \in \left\{ 0,\ 1,\ \cdots,\ 2^k \right\} \qquad (m_1)$$

$$s_2 \in \left\{ 0,\ 1,\ \cdots,\ 2^k - 1 \right\} \qquad (m_2)$$

$$s_3 \in \left\{ 0,\ 1,\ \cdots,\ 2^k - 2 \right\} \qquad (m_3)$$

be the incoming symmetrical residues for which we desire the corresponding value (in binary). Although there is a direct correspondence between the OSNS and the RNS, these values cannot be converted in a straight forward manner, e.g., using the Chinese Remainder Theorem (CRT).

Let h be an integer corresponding to the unknown $0 \le h \le M - 1$ where M represents the dynamic range of the three moduli OSNS described by Eq. (3). By definition of the OSNS we know that either

$$h \equiv s_i \left(\bmod\ 2m_i\right) \qquad (4)$$

or

$$h \equiv 2m_i - s_i - 1 \pmod{2m_i}, \tag{5}$$

for $i = 1, 2$ and 3. This yields 2^3 systems of linear congruence equations of which only two have a solution. Because these congruences are with respect to $2m_i$ and the greatest common devisor of $2m_i$ and $2m_j$ is 2 ($i \neq j$), it follows from the generalization of the CRT that, for the two sets of congruences that have a solution either all of the remainders must be odd or all of the remainders must be even [Ref. 8]. For example $k = 6$, ($m_1 = 65$, $m_2 = 64$, $m_3 = 63$), and $s_1 = 2$, $s_2 = 1$, and $s_3 = 0$. Then

$$h \equiv 2 \pmod{130} \qquad \text{or} \qquad h \equiv 125 \pmod{130}$$

and

$$h \equiv 1 \pmod{128} \qquad \text{or} \qquad h \equiv 126 \pmod{128}$$

and

$$h \equiv 0 \pmod{126} \qquad \text{or} \qquad h \equiv 125 \pmod{126}$$

The only congruences with solution are

$$\begin{aligned} h &\equiv 2 \pmod{130} \\ h &\equiv 126 \pmod{128} \\ h &\equiv 0 \pmod{126} \end{aligned} \tag{6}$$

and

$$\begin{aligned} h &\equiv 125 \pmod{130} \\ h &\equiv 1 \pmod{128} \\ h &\equiv 125 \pmod{126} \end{aligned} \tag{7}$$

By the same theorem, these two sets of congruences each have a unique solution (mod $2M$). For the even remainders h will be even and we can divide the entire congruence by 2. This is a standard CRT problem. For example from Eq. (6)

$$\begin{aligned} \frac{h}{2} &\equiv 1 \pmod{65} \\ \frac{h}{2} &\equiv 63 \pmod{64} \\ \frac{h}{2} &\equiv 0 \pmod{63} \end{aligned} \tag{8}$$

The solution of the odd remainder case can be obtained directly from the even remainder case. In fact, if h_0 solves the even remainder case, then $2M - h_0 - 1$ solves the odd remainder case. We know that exactly one of these values lies within the dynamic range M and is the desired value of h. Note that given s_1, s_2 and s_3 it is not clear which of the two solutions will fall within the dynamic range of the system. However, because of the solution process given, it does not matter.

The block diagram of the design used to implement this algorithm is shown in Figure 12. The inclusion of edge-triggered flip-flop registers (1-bit, 7-bit and 13-bit latch) allows for the design to be pipelined which allows a three-fold increase in the conversion speed. The conversion processes is accomplished in three steps; these are shown in Figure 12 as the following: OSNS to RNS, RNS to Binary and RNS to OSNS. The steps of the conversion process follow.

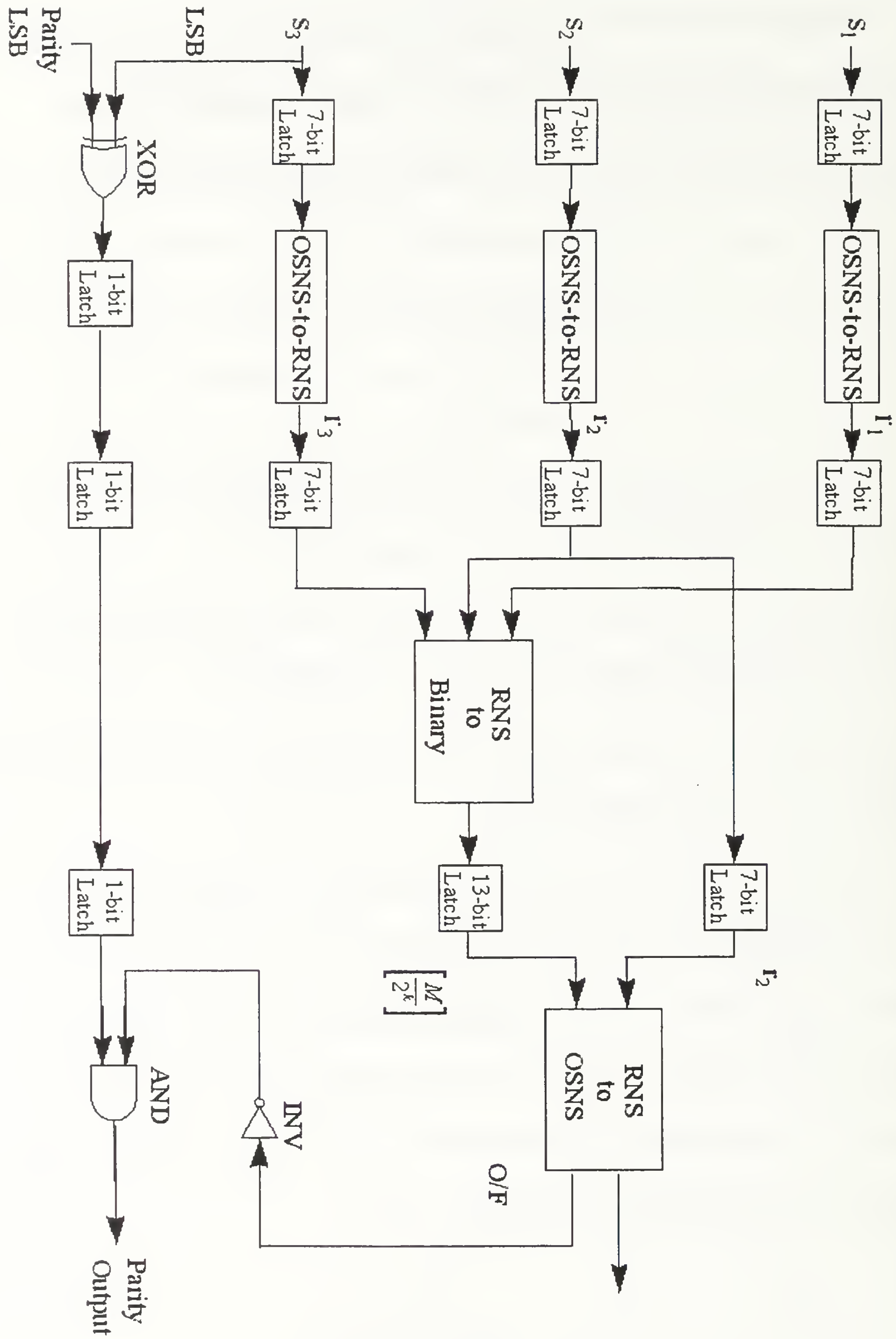


Figure 12 OSNS to binary conversion.

B. OSNS RESIDUES TO RNS RESIDUES

For simplicity let $n = h/2$. Reference 5 provides the solution for

$$\begin{aligned} n &\equiv r_1 \pmod{m_1} \\ n &\equiv r_2 \pmod{m_2} \\ n &\equiv r_3 \pmod{m_3} \end{aligned} \tag{9}$$

where

$$r_i = \begin{cases} \dfrac{s_i}{2} & \text{for } s_i \text{ even} \\ \dfrac{2m_i - s_i - 1}{2} & \text{for } s_i \text{ odd.} \end{cases}$$

The block diagram of the design used to accomplish this algorithm is shown in Figure 13. The diagram shown is for the mod 65 (s_1) channel and is typical of the designs used for the mod 64 and mod 63 residues. It is important note that $2m_i - s_i - 1$ is the same as ones complement subtraction of s_i from $2m_i$. For this operation the decision is based on the least significant bit (LSB) of the incoming data. If the LSB is 0 then the incoming bits are right shifted one position and become RNSA out. If the LSB is 1 then the input is inverted and added to $2m_1$ (130, bit pattern 10000010). This performs the ones complement subtraction. The result is then right shifted and becomes RNSA out. The LSB of the incoming data controls the multiplexers at the right hand side of the schematic which select either the even case (LSB = 0) or the odd case (LSB = 1).

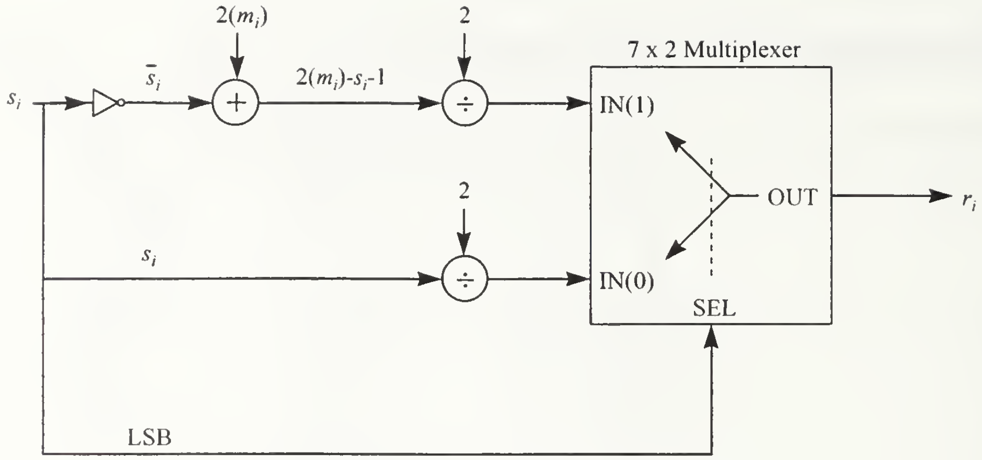


Figure 13 SNS to RNS conversion (typical).

C. RNS RESIDUES TO RNS-BASED BINARY

Following the algorithm presented in Reference 5 and letting

$$r_i = b_{ik} b_{i(k-1)} \cdots b_{i0},$$

the algorithm for converting the RNS residues to binary output is summed up in the following two equations

$$n = \left[\frac{n}{2^k}\right] 2^k + r_2 \qquad (10)$$

and

$$\left[\frac{n}{2^k}\right] = \left|(C - r_1) + (B + A)\right|_{2^{2k}-1} \qquad (11)$$

where [ ] denotes the integer part of the operand and $|m|_{2^{2k}-1}$ represents the residue of m modulo $2^{2k}-1$, an integer in $[0, 2^{2k}-2]$. Here

$$C = b_x b_{1(k-1)} \cdots b_{12} b_{11} b_x b_{1((k-1)} \cdots b_{11} \qquad (12)$$

and $b_X = b_{10} \; V \; b_{1k}$ (V denotes logic OR operation). (The circuit for this operation is shown in Figure 14.) Also,

$$B = \bar{b}_{2(k-1)}\bar{b}_{2(k-2)} \cdots \bar{b}_{22}\bar{b}_{21}\bar{b}_{20}1 \cdots 1 \tag{13}$$

as shown in Figure 15, and

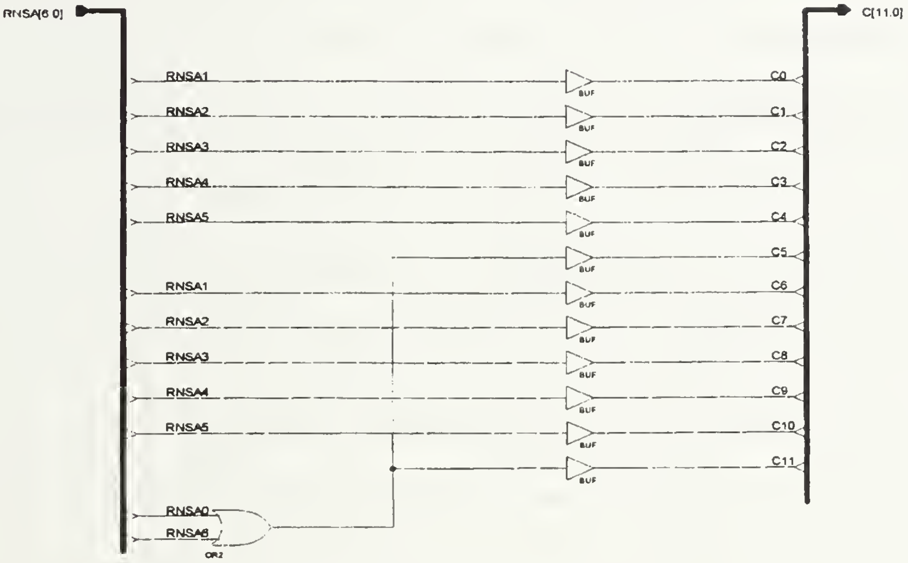


Figure 14 RNSA to C conversion

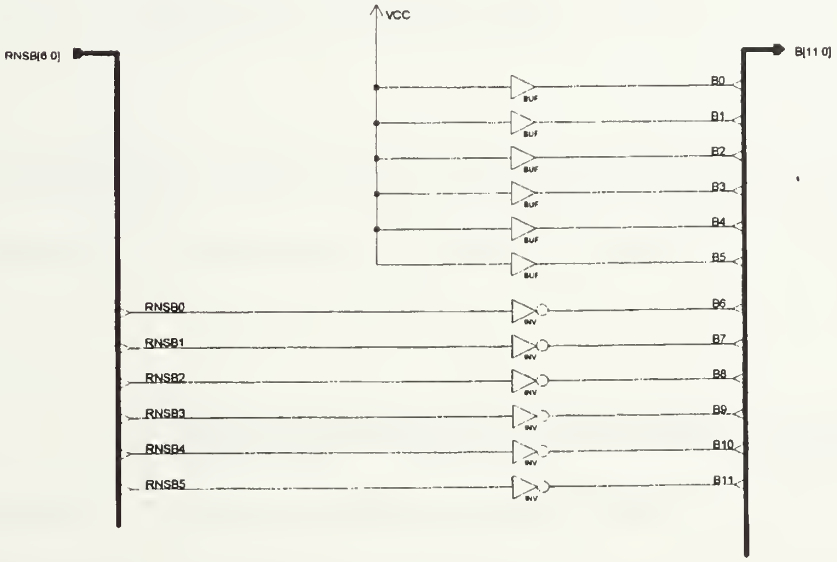


Figure 15 RNS B to B conversion.

$$A = b_{30}b_{3(k-1)} \cdots b_{32}b_{31}b_{30}b_{3(k-1)} \cdots b_{32}b_{31} \tag{14}$$

as shown in Figure 16.

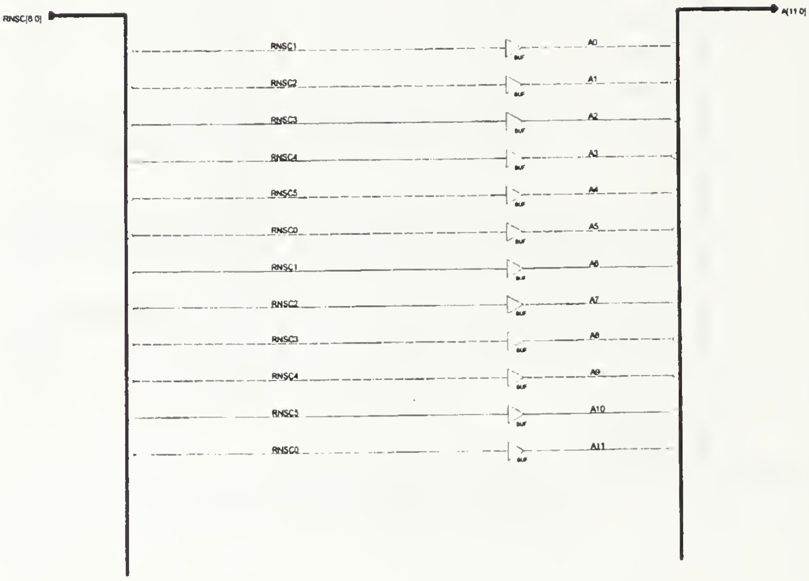


Figure 16 RNSC to A conversion.

The rest of the conversion process from RNS to binary corresponding to Eq. (11) is shown in Figure 17. This stage produces the 12 most significant bits of the RNS based binary output.

D. RNS BASED BINARY TO OSNS BASED BINARY

The last step in the OSNS-to-binary conversion completes Eq. (10) and produces the RNS based binary number (n). The algorithm for conversion from the RNS based binary to OSNS based binary is or

$$\text{if } n < M/2 \qquad h = 2n \text{ (left shift the bits of } h \text{ one position).}$$

The block schematic diagram of this operation is shown in Figure 18. This diagram also shows the ones complement subtraction for the 11 most significant bits of the RNS residues. Since the lower seven bits of $2M$ are 0, the corresponding bits in the RNS residue are just inverted and routed to the D1 ports on the 2-bit multiplexers (M2_1). The logic circuit consisting of the AND and OR gates in the upper left portion of the diagram determine if the RNS based binary input is greater than $M/2$ (where the "less than" condition selects the D0 ports on the multiplexers and the "greater than or equal to" condition selects the D1 ports). The 4-bit OR gate in the upper right section of the schematic is used to determine if the SNS based binary output is greater than 14 bits and is used in the parity circuitry described in the next section. The full set of schematics for the FPGA implementation of this design are located in Appendix D.

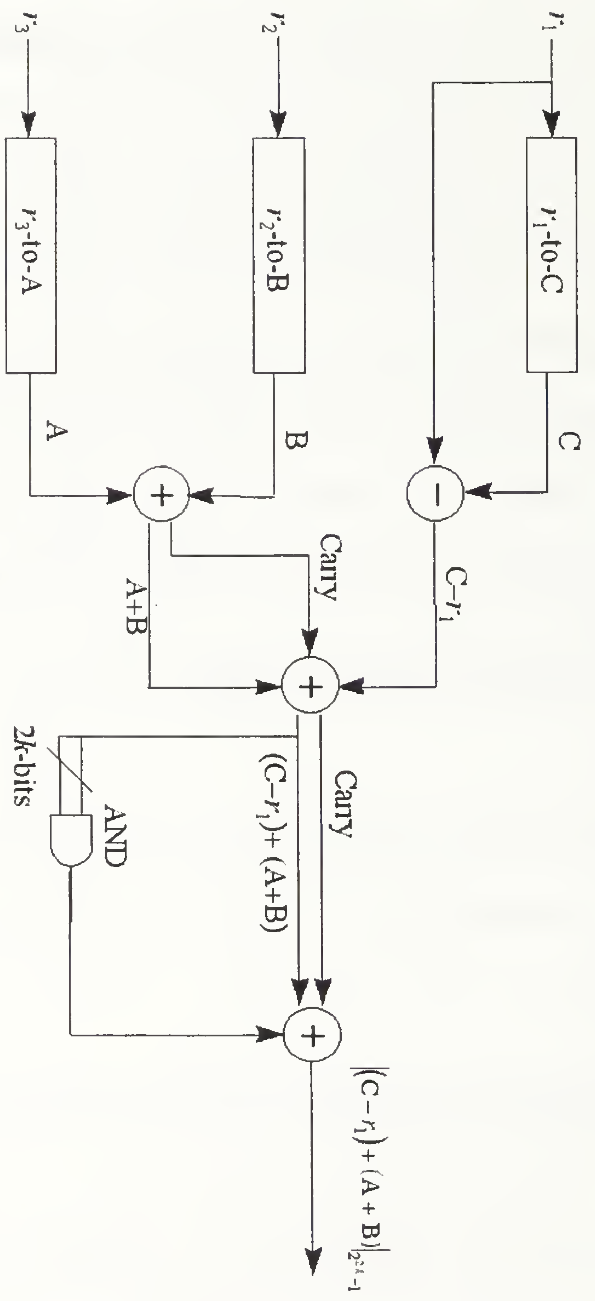


Figure 17 RNS to binary conversion Eq. (11).

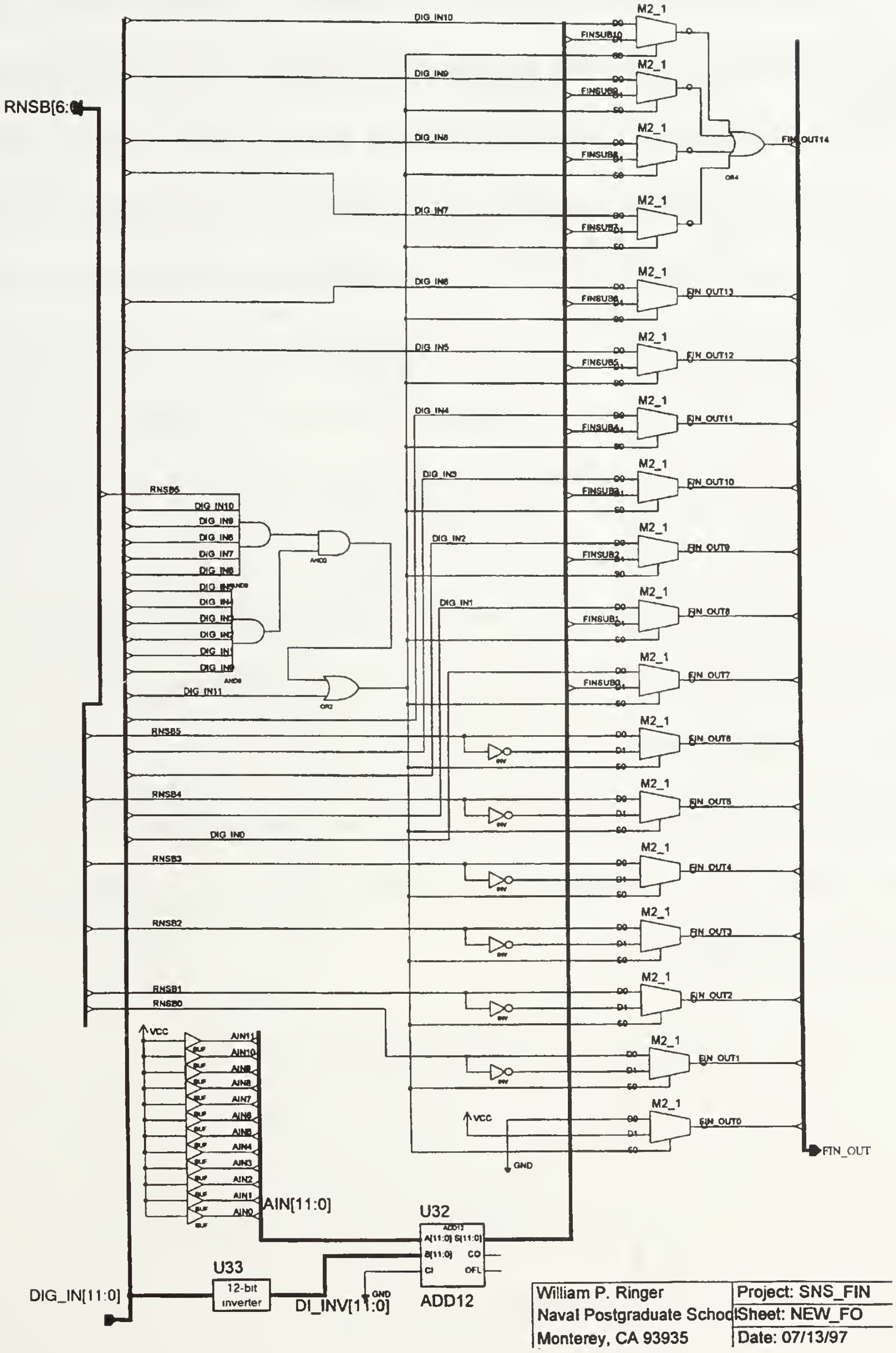


Figure 18 SNS to binary final processing.

E. PARITY PROCESSING

The parity processing, as described in Chapter II, is initiated on the schematic depicted in Figure 12. In the lower left section of the schematic the least significant bits of both the MOD 63 input and the PARITY input are EXCLUSIVE ORed together to produce the parity bit. This bit is pipelined through three flip-flops and then ANDed with the inverted output of the FIN_OUT bit 14. This ensures that, if either the parity is even or the result is greater than the 14 -bit range of the system, the parity output of the OSNS FPGA will be 0. That is, a 1 output on this line indicates a good output for the FPGA.

The circuit depicting the design of the parity processing FPGA is shown in Figure 19. Components U3 and U4 are 14-by-2 bit multiplexers. Components U1 and U2 are 14-bit edge triggered flip-flop registers used for storing partial results. Multiplexer U4 selects, based on the parity input signal, either the 14-bit result from the OSNS FPGA or the last known good 14-bit result routed from the output of register U1. Multiplexer U3 selects, based on the parity input signal, either the last known good output from register U1 or the 14-bit input from the OSNS FPGA. Multiplex U2 is used to register the output of the parity processing FPGA.

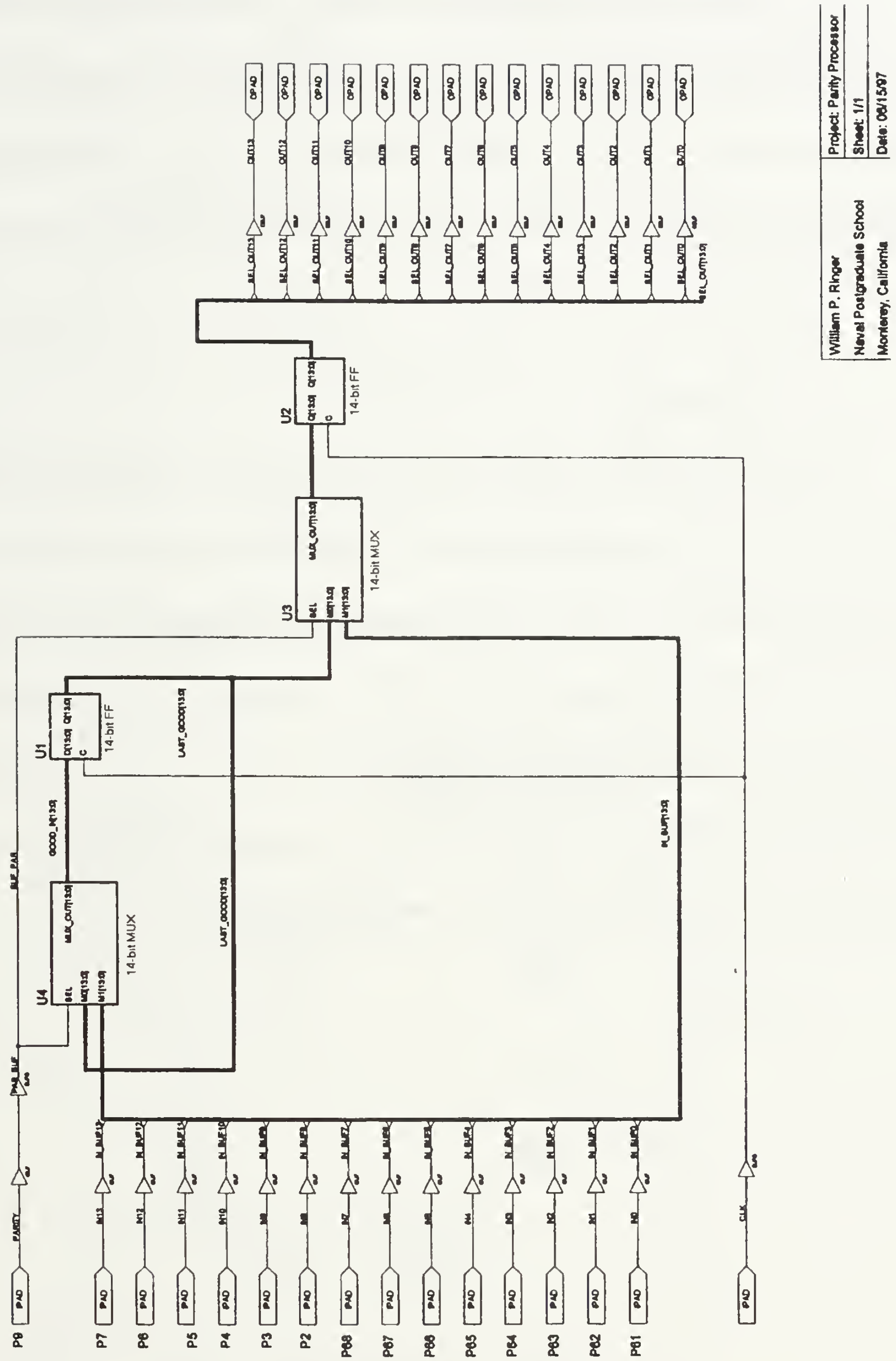


Figure 19 Parity processing FPGA.

F. FPGA DETAILS

The FPGAs used to implement this design are the XILINX XC4003A for the OSNS design and XC3020A for the parity processing design. Both of the FPGA designs were created using Foundation ActiveCAD created by ALDEDC. The designs were then translated into program bitstreams by XACTSTEP6 XILINX's FPGA programming tool.

1. XC4003A

The XILINX XC4003A is one of the third generation of XILINX FPGA products. It uses a sub-micron Complimentary Metal Oxide Semiconductor (CMOS) design and has an approximate gate count of 3000 devices. The major processing component in the FPGA is the Configurable Logic Block (CLB). A diagram of a typical XC4000A class CLB is shown in Figure 20. The XC4003A has 100 of these CLBs. The XC4003A also has 360 flip-flop which are used in the design to pipeline the data from the individual stages and 80 input/output (IOB) blocks as shown in Figure 21.

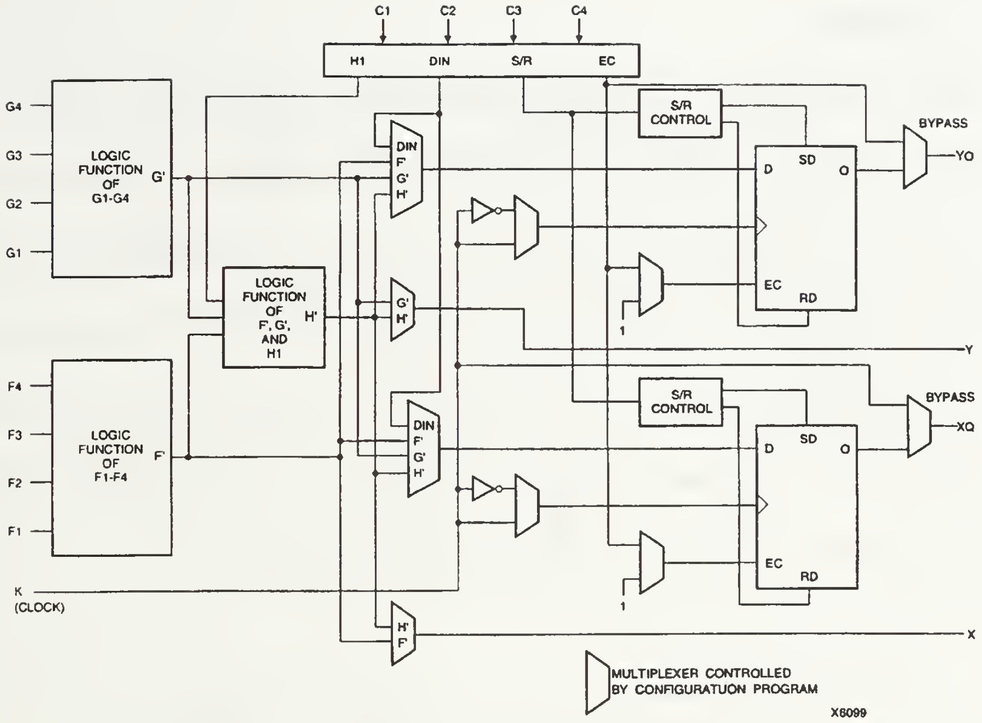


Figure 20 XC4003A CLB. From Ref. [7].

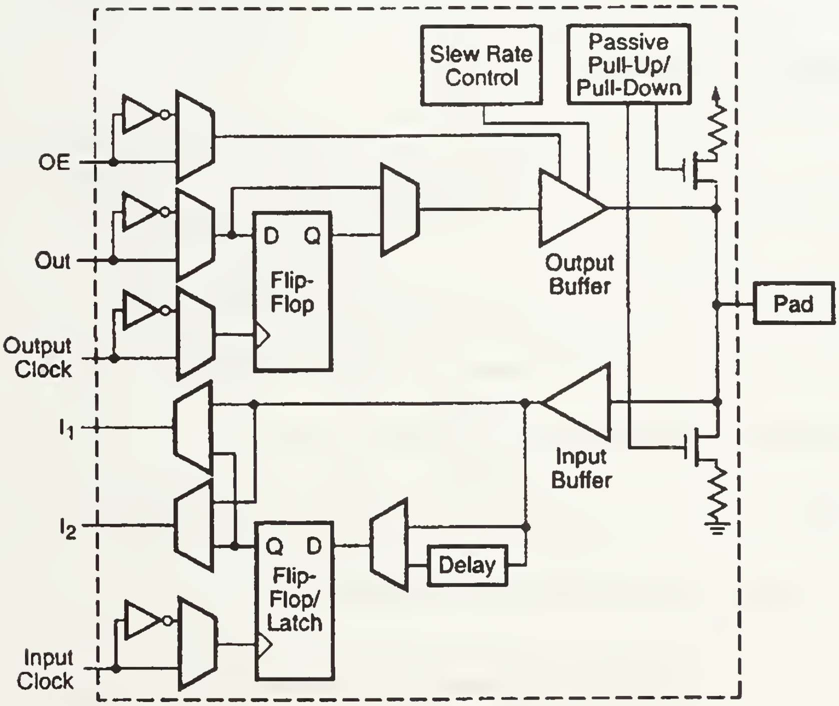


Figure 21 XC4003A IOB. From Ref. [7]

2. XC3020A

The CMOS XC3000 class of FPGA family provides high density, high performance digital integrated circuits. The XC3020A consists of 64 configurable logic blocks (Figure 22) which are configured in an 8 x 8 matrix formation. It also contains 256 flip-flops for pipelining operations. Sixty-four user input/output blocks are available (Figure 23). This FPGA was chosen for its small size and because the test and evaluation board (described in the next section) was delivered with an XC4003A and an XC3020A installed which allowed for immediate integration into the project.

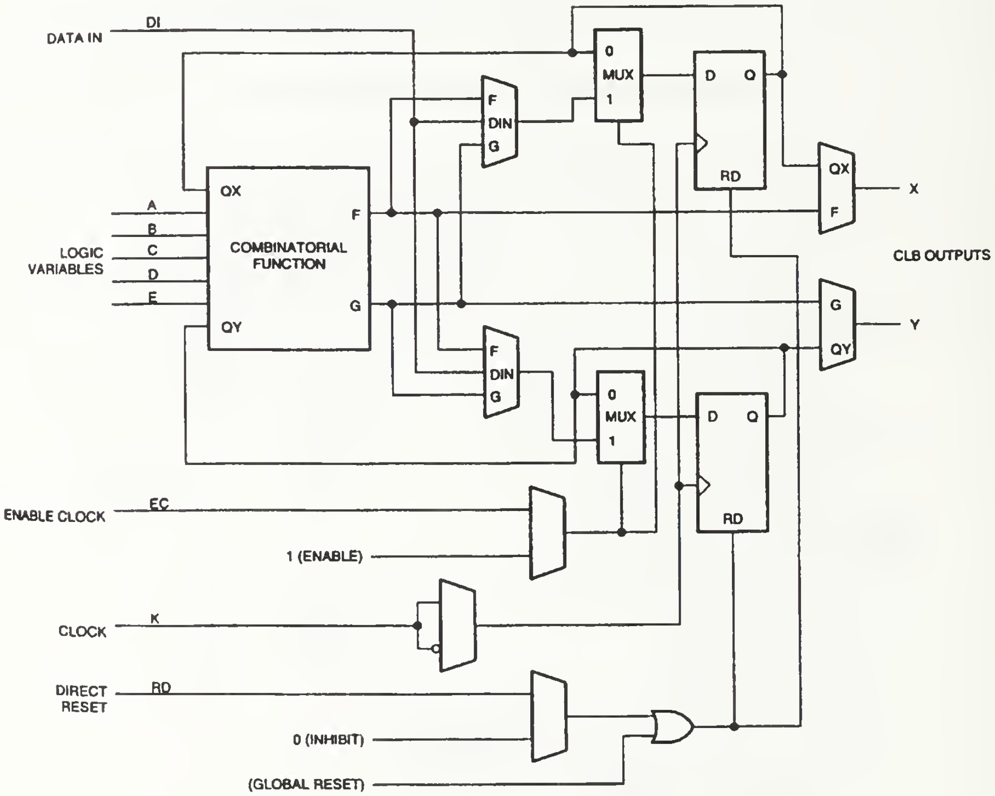


Figure 22 XC3020A CLB. From Ref. [7].

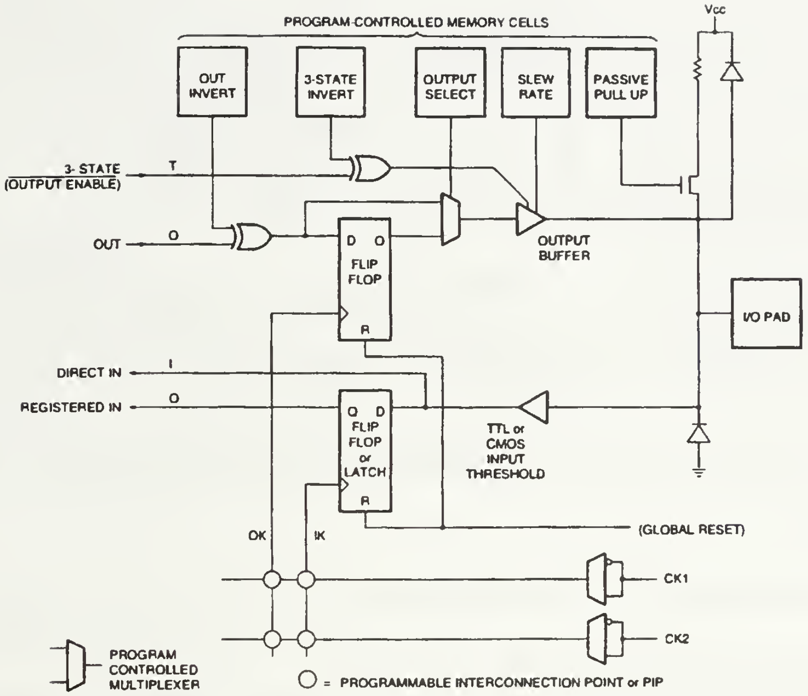


Figure 23 XC3020A IOB. From Ref. [7].

3. Test And Evaluation Board

Programming, testing and system integration of the FPGAs was accomplished using the XILINX FPGA Demonstration Board shown in Figure 24. This XC4003A and XC3020A are clearly shown in the figure. Around the FPGAs are interconnect pins (headers) which allow any of the input/output ports on the FPGAs to be monitored and connected to external equipment.

The FPGA demonstration board comes with the following features:

- One socket for an XC3000 device
- One socket for an XC4000 device

- One 17XXX socket for each FPGA (used for programming the FPGA)
- An XChecker/Download cable header for each FPGA
- 8 DIP switches to set up the XC4000 and XC3000 FPGAs
- 16 interconnect lines that connect the two FPGAs
- 8 DIP switches which set logic input levels; switch outputs, which drive both FPGAs; closing switches, which drive signals to logic 1's
- Program, Reset and Spare push-button switches which are common to both FPGAs
- XC3000 displays that use eight LED bars in one row and one 7-segment LED
- XC4000 displays that use eight LRD bars in one row and two 7-segment LEDs
- Space for an optional +5 V regulator for battery operation
- Prototype area on PC board.

Figure 25 shows the schematic of the FPGA Demonstration Board. Descriptions of the major components follows. All information is from Reference 7.

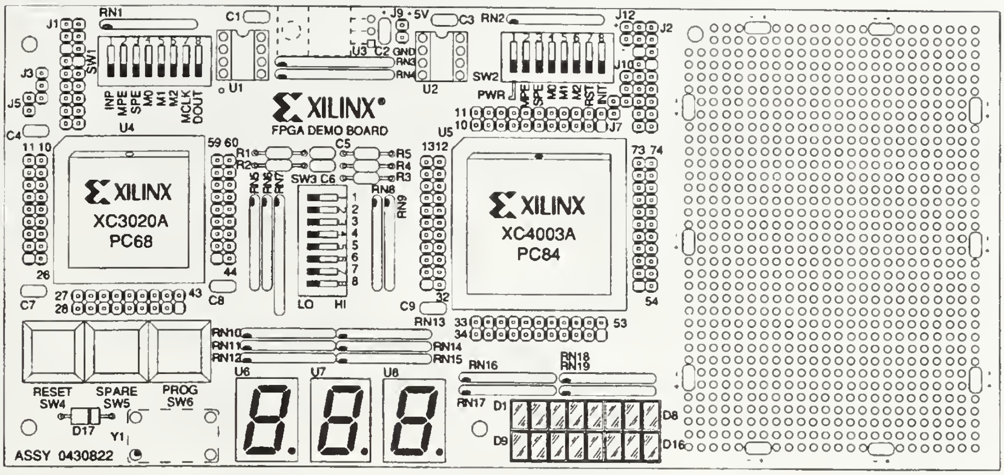


Figure 24 XILINX FPGA demonstration board. From Ref. [7].

a. +5 V Power Connector (J9)

A regulated +5 volts and ground connected to the FPGA Demonstration Board through J9. Pin 1 (square pad) is +5 V and pin 2 is ground. The power supply should provide at least 250 mA of current to drive the LED displays.

b. Unregulated Power Input (J12)

The unregulated power input provides a way to power the FPGA Demonstration Board from an unregulated source such as a 9 V battery or an a.c. adapter. The input should be 7 VDC - 12 VDC at 250 mA, typically. Consideration must be given to the power dissipation requirements of the U3 voltage regulator if the voltage input is greater than 9 V.

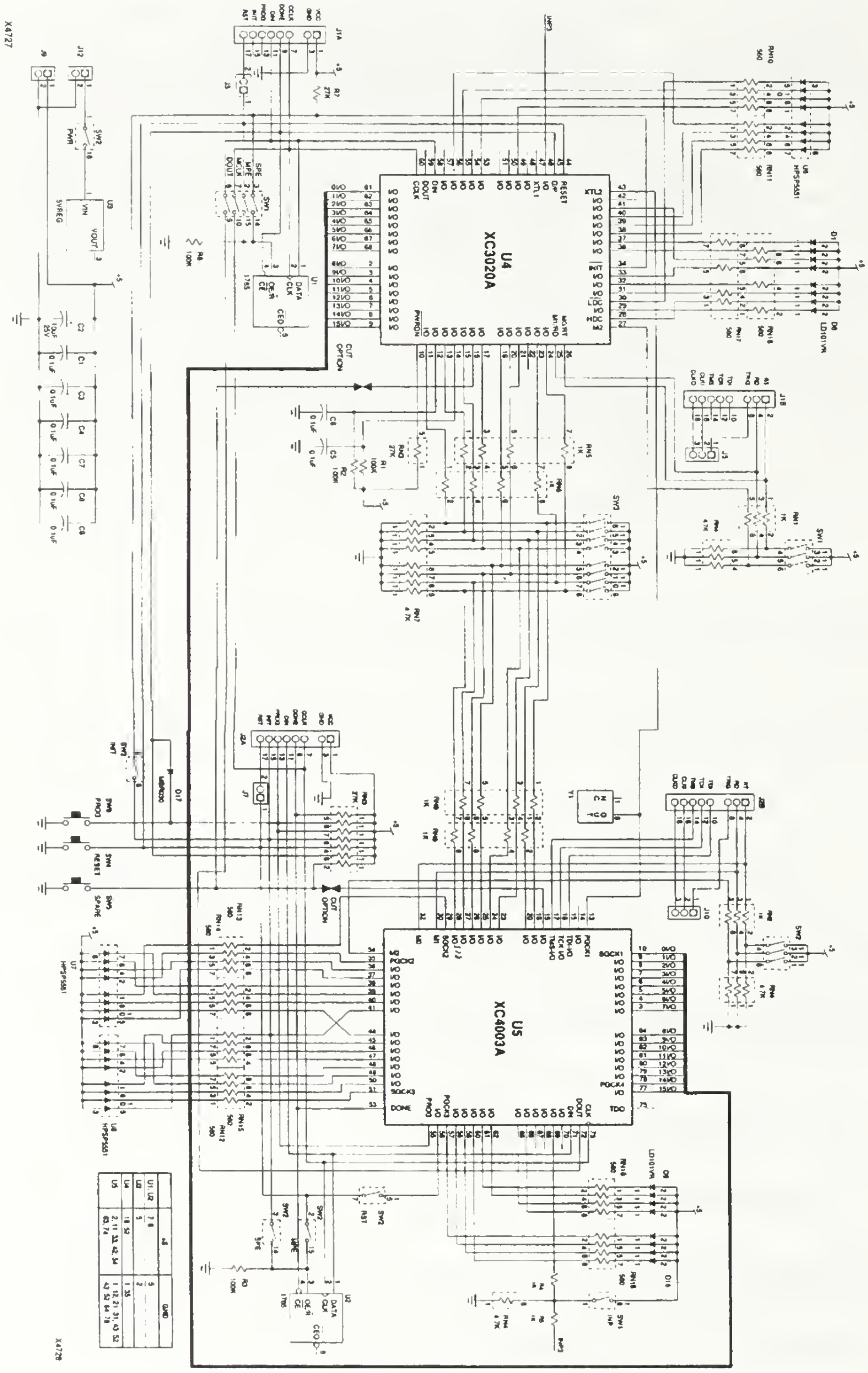


Figure 25 FPGA Demonstration Board schematic. From Ref. [7].

c. +5 V Regulator Option (U3)

A three terminal +5 V regulator, such as the LM2940CT (used in this project), to power the demonstration board from an unregulated power supply, such as a +9 V battery. Pin 1 (square pad) is V_{in}, pin 2 is ground and pin 3 is +5 V out.

d. RESET Push-button (SW4)

When pressed, the RESET push-button can apply an active-Low Reset signal to the FPGAs depending on how the Reset signal routing is configured. Reset is normally pulled High through a 27-kΩ resistor.

e. SPARE Push-button (SW5)

The SPARE push-button applies an active-Low signal to the XC3020A pin 16 and to the XC4003A pin 18. These pins can be isolated from the switch by using the trace-cut options on the solder side of the board. The SPARE signal is pulled High through a 27-kΩ resistor.

f. PROG Push-button (SW6)

The PROG push-button applies an active-Low signal to the DONE/PROGRAM input of the XC3020A FPGA socket and to the PROGRAM input on the XC4003A FPGA socket. The PROG signal is normally pulled high through a 13.5-kΩ resistor.

g. Eight General Purpose Input Switches (SW3)

These eight switches connect to eight general-purpose inputs on both the XC3020A and the XC4003A FPGAs. These switches provide logic inputs to the FPGAs. An FPGA input pin is set to a logic 1 when a switch is on, and a logic 0 when a switch is off.

h. 7-Segment Displays (U6, U7, U8)

Three 7-segment displays are included. The leftmost display (U6) is connected to the XC3020A, and the right two displays (U7 and U8) are connected to the XC4003A.

Each LED is turned on by driving the corresponding FPGA pin LOW with a logic 0. The decimal point on U8 connects to the INIT pin of the XC4003A and serves as a programming error indicator. The decimal point should be on while the FPGA is in its initial clearing state, then it should remain off during configuration. If the decimal point comes back on, there has been a programming error.

The decimal points on U6 and U7 are connected to the LDC (Low during configuration) pins of the XC3020A and XC4003A, respectively. The decimal points are on while the FPGAs wait to be configured.

i. LED Indicators (D1-D8, D9-D16)

Eight LEDs connect to the I/O pins of each FPGA. D1 through D8 connect to the XC3020A and D9 through D16 connect to the XC4003A. Each LED can be turned on by driving its corresponding FPGA pin LOW with a logic 0.

j. I/O Line Connectors

There are 16 I/O lines that connect the XC3020A and XC4003A FPGAs, as shown in Table 3. For this project these are the lines used to route the 14-bit data output and the 1-bit parity output from the XC4003A to the XC 3020A.

k. Optional Crystal Oscillator (Y1)

A standard 4-pin crystal oscillator can be added to the FPGA Demonstration board. The oscillator output drives the XC3020A XTL2 input and the XC4003A PGCK1 input.

I/O Line	XC3020A Pin	XC4003A Pin
0	61	10
1	62	9
2	63	8
3	64	7
4	65	6
5	66	5
6	67	4
7	68	3
8	2	84
9	3	83
10	4	82
11	5	81
12	6	80
13	7	79
14	8	78
15	9	77

Table 3 I/O line connections for the XC3020A and XC4003A devices. From Ref. [7].

l. Prototype Area

The prototype area is a 0.1-inch grid of holes where additional circuitry can be added to the demonstration board. A +5 V bus (component side) and a ground bus (solder side) are available on the perimeter of this area. There are also locations for filter capacitors.

m. XC4003A FPGA and Socket (U5)

The XC4003A occupies U5 on the demonstration board.

n. XC4003A Probe Points

All pins on the XC4003A connect to the headers that surround the FPGA socket. These pins provide convenient points for probing signals or making wirewrap connections to external circuitry, such as on the prototype area. Pin numbering increases from the inside row to the outside, counterclockwise. The corners of each header lists the starting number for that header.

o. XC4003A Configuration Switches (SW2)

The following are descriptions of the SW2 switches:

- PWR - Power (SW2-1)

This switch turns the unregulated power input on or off to the +5 V regulator U3.

- MPE - Multiple Program Enable (SW2-2)

With MPE Turned on and SPE turned off, the configuration PROM (U2) is reset by the RESET push-button (SW4). Configuration mode must first be set to master-serial. After a Reset or power-up, the first bitstream stored in the serial PROM is loaded into the XC4003A. Pressing RESET resets the serial PROM address pointer. Pressing PROG (SW6) loads the XC4003A with the first bitstream again. If PROG is pressed without first pressing RESET, the XC4003A is loaded with the next bitstream that is stored in the serial PROM. The size of the serial PROM limits the number of bitstreams that can be sequentially loaded.

- SPE - Single Program Enable (SW2-3)

With SPE turned on and MPE turned off, the configuration PROM (U2) is reset by the XC4003A's INIT output, which is driven Low whenever PROG (SW6) is pressed. The first bitstream stored in the serial PROM is loaded into the XC4003A.

- M0, M1, M2 - Mode Pins (SW2-4,5,6)

These three switches must be on to configure the XC4003A using the XChecker/Download Cable. When these switches are on, the FPGA is in slave serial mode. To configure the XC4003A from the onboard serial PROM, these three switches must be off, placing the FPGA in master-serial mode.

- RST - Reset (SW2-7)

When this switch is on, it connects the RESET push-button (SW-4) to the XC4003A I/O pin 56.

- INIT - Initialize (SW2-8)

When this switch is on, it connects the XC3020A INIT pin to the XC4003A INIT pin. This connection is used to configure the FPGAs in a daisy chain with the XC4003A at the head of the chain.

p. XC4003A XChecker/Download Cable Connector (J2)

This connector allows for the connection of the XChecker/Download cable to the XC4003A for programming. A description of each of the pins of the XChecker/Download cable is given in Reference 7.

With the Download Cable connected, J2-9 provides both the DONE and PROG functions. Since the XC4003A requires a program input that is separate from DONE, the PROG push-button must be pressed before configuring the XC4003A.

q. Jumper J7 and Tiepoints J10 (1-3)

Jumper J7 allows the XChecker signal RST on J2-17 to drive the reset line on the demonstration board. Tiepoint pins jumper the following XChecker signals to the circuit. Tiepoint J10-1 connects to TRIG on J2-6, Tiepoint J10-2 connects to CLK1 on J2-16, and Tiepoint J10-3 connects to CLK0 on J2-18.

r. Serial PROM Socket (U2)

This serial PROM configures the XC4003A or the XC4003A and the XC3020A connected in a daisy chain. The configuration mode must be “master-serial” to configure the FPGAs from the serial PROM; however, the serial PROM was not used for this project.

s. XC3020A FPGA and Socket (U4)

The XC3020A FPGA occupies socket U4 on the demonstration board.

t. XC3020A Probe Points

All pins on the XC3020A FPGA connect to the headers that surround the FPGA socket. These pins provide convenient points for probing signals or making wirewrap connections to external circuitry, such as the prototype area. Pin numbering increases from the inside row to the outside, counterclockwise. The corners of the headers list the starting numbers for that header.

The XC3020A I/O pins 2 through 9 and 61 through 68 connect to XC4003A pins 3 through 10 and 77 through 84, respectively. The XC3020A pins share the XC4003A probe points header.

u. XC3020A Configuration Switches (SW1)

The following are discussions of the SW1 switches:

- INP - Input Switch (SW1-1)

This extra switch is connected to provide an extra logic to the XC3020A pin 46 and the XC4003A pin 69. The FPGA input pins are set to a logic 1 when the switch is in and a logic 0 when the switch is off.
The FPGA pins connected to this switch are intended to be used as inputs; however, they have a 1 kΩ resistor that isolates them from the switch, so it is possible to define them as output. It is also possible to drive the them from an external source by connecting that signal to the FPGA probe point header.

- MPE - Multiple Program Enable (SW1-2)

When MPE is on and SPE is off, the configuration PROM (U1) is reset by the RESET push-button (SW4). Configuration mode must be set to master-serial. After a Reset or power-up, the first bitstream stored in the serial PROM is loaded into the XC3020A FPGA. If the RESET push-button is pressed, the serial PROM address pointer is reset. If the PROG (SW6) push-button is pressed, the XC3020A is loaded with the first bitstream again. If the PROG push-button is pressed without pressing RESET, the XC3020A is loaded with the next bitstream stored in the serial PROM. The number of bitstreams that can be sequentially loaded is limited by the size of the serial PROM.

- SPE - Single Program Enable (SW1-3)

When SPE is on and MPE is off, the configuration PROM (U1) is reset by the XC3020A's INIT output, which is driven Low whenever the PROG push-button is pressed. The first bitstream stored in the serial PROM is loaded into the XC3020A FPGA.

- M0, M1, M2 - Mode Pins (SW1-4,5,6)

To configure the XC3020A using the XChecker/Download Cable, these switches must be on, placing the XC3020A FPGA in slave-serial mode. To configure from the onboard serial PROM, these switches must off to place the FPGA in the master-serial mode.

- MCLK - Master Clock (SW1-7)

When this switch is on, it connects the XC4003A configuration clock (pin 73) to the configuration clock on the XC3020A (pin 60). This connection is used to configure the FPGAs in a daisy chain with the XC400A at the head.

- DOUT - Data Out (SW1-8)

When this switch is on, it connects the XC4003A data out line (pin 72) to the data in line of the XC3020A. This connection configures the FPGAs in a daisy chain with the XC4003A at the head.

w. Serial PROM Socket (U1)

This serial PROM is used to configure the XC3020A. The XC3020A must be in the master-serial mode to configure from the serial PROM.

This chapter has discussed the design of the OSNS and parity FPGAs and introduced the FPGA demonstration board used to realize these designs. The next chapter will explore the timing aspects of the digital antenna project.

V. TIMING

Timing is an essential aspect of this design. The synchronization of the ODECL latching outputs to the comparator board output latches is critical. The ODECL waveforms must trigger the output latches of the comparator boards within the 6 ns that the pulse from the d.c. restoration circuit is at the latches.

The OSNS and parity FPGAs must also be clocked at the proper time to ensure continuity of the pipelined data out of the comparator boards. That is, enough time must be given for the comparator board outputs to stabilize prior to triggering the input latching registers on the OSNS FPGA . This stabilization time is called the setup time for the input flip-flop latches on the FPGAs. If the FPGA is triggered prior to the setup time, the data latched into the flip-flops will be unreliable. The same is true of the output data from the OSNS FPGA and U1 on the parity processing FPGA. A block diagram of the timing interconnect is shown in Figure 26.

The data output from the HP-71600B Bit Error Encoder (6 ns pulse width and 200 ns pulse repetition time) is routed to the BCP-400 Laser Transmitter and through the optical subsection as shown in Figure 2. The sync output of the HP-71600B is routed to the trigger input of a Wavetek-145 Function Generator which is set to trigger on the HP-71600B input. The TTL output of the Wavetek-145 provides the clock input for both the OSNS and parity FPGAs.

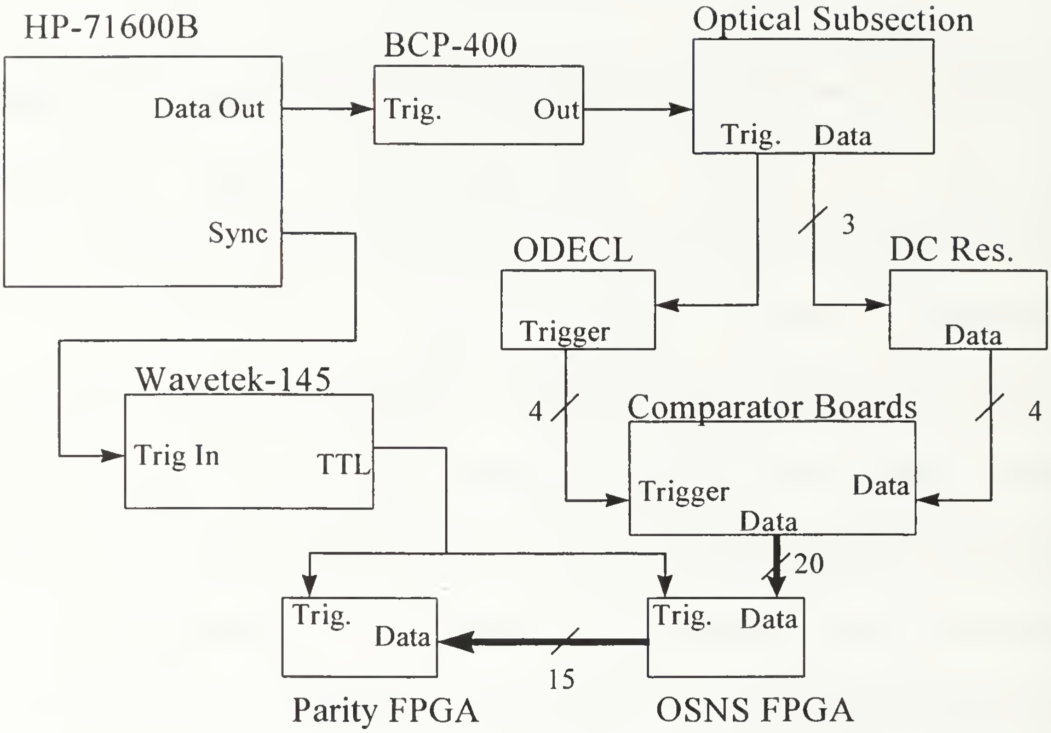


Figure 26 Trigger diagram.

This concludes the discussion of the processing stages of the design. The next chapter will discuss the outputs of the separate subsections.

VI. OUTPUTS

A. OPTICAL SUBSECTION

A typical output from the optical subsection is shown in Figure 27. This figure shows the reference pulse out of the optical receiver, used to initiate the ODECL circuitry, and the data pulse outputs from the d.c. restoration circuitry for mod-65, 64 and 63. The parity data pulse output from the d.c. restoration circuitry is not shown but is the same as the mod-63.

The data pulses are routed to the comparator boards where they are quantized as discussed in Chapter III. The output of the optical receiver is routed to the input of the ODECL trigger circuitry where it is processed as described in Reference 3. The outputs of the ODECL trigger circuitry are routed to the output latches of the comparator boards as described in Chapter III.

The noise apparent on the data pulses has been measured and, using the formula

$$SNR = 20 \log \left(\frac{V_{noise}}{V_{pulse}} \right) \qquad (15)$$

to calculate the signal-to-noise ratio (SNR), the following results were achieved: SNR(mod63) = 23.63 dB, SNR(mod64) = 23.23 dB and SNR(mod65) = 13.064 dB. These results will be discussed further in the conclusions chapter.

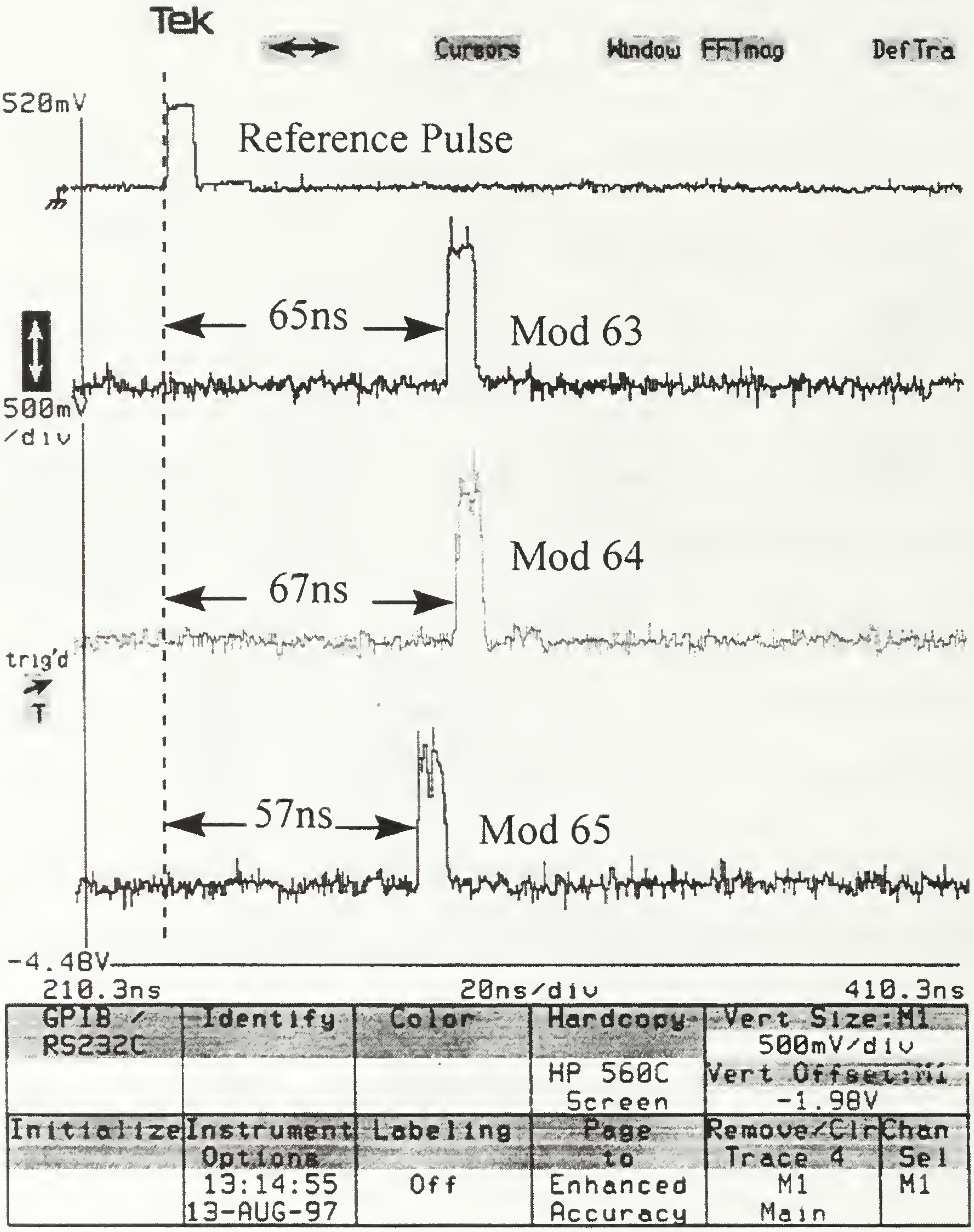


Figure 27 Optical subsection outputs (typical).

B. ODECL OUTPUT

The output of the ODECL circuitry for one of the channels is shown in Figure 28. The output latches on the comparator boards will latch the data levels when the ODECL triggers cross at 0.6 volts. One of the interesting aspects of the comparator boards is that the latches latch any time the triggers cross at 0.6 volts, so care has to be taken that the trailing edges of the trigger pulses are aligned to the incoming data pulses. (This is shown as a vertical line in Figure 28.) If not, the data output of the comparator boards is good for 30 ns and then noise is latched onto the outputs.

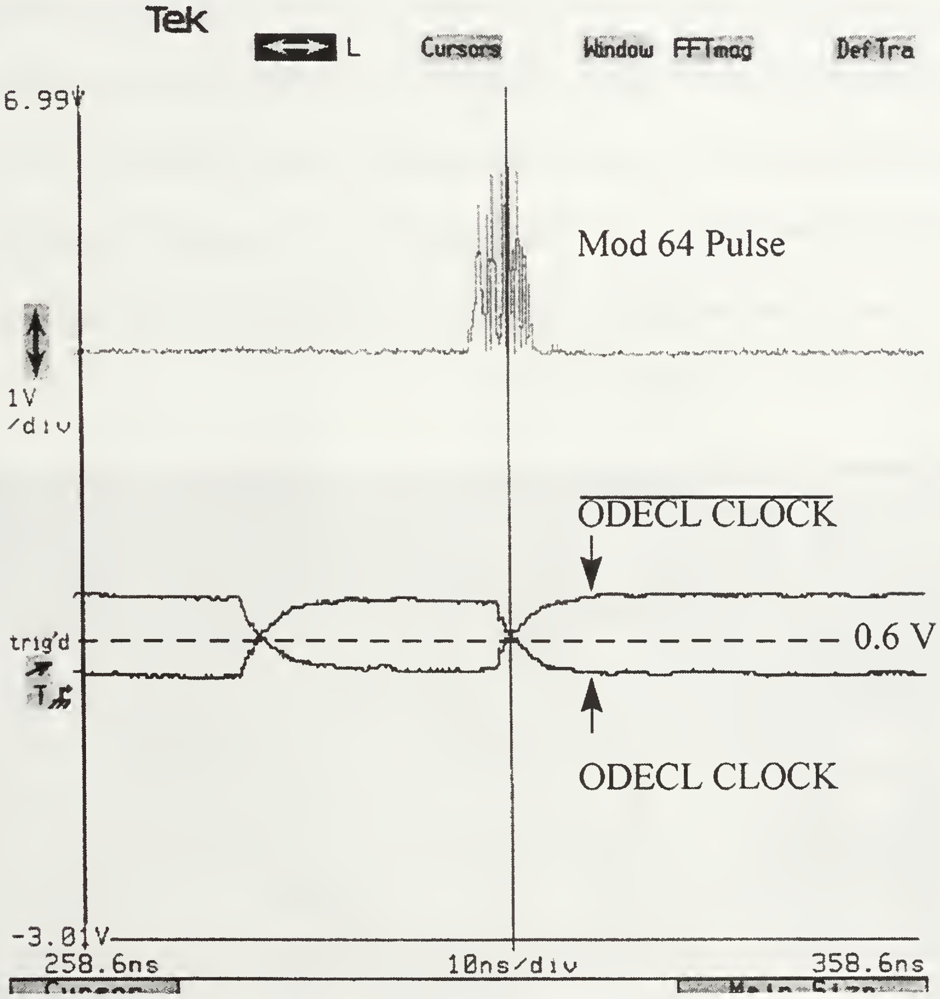
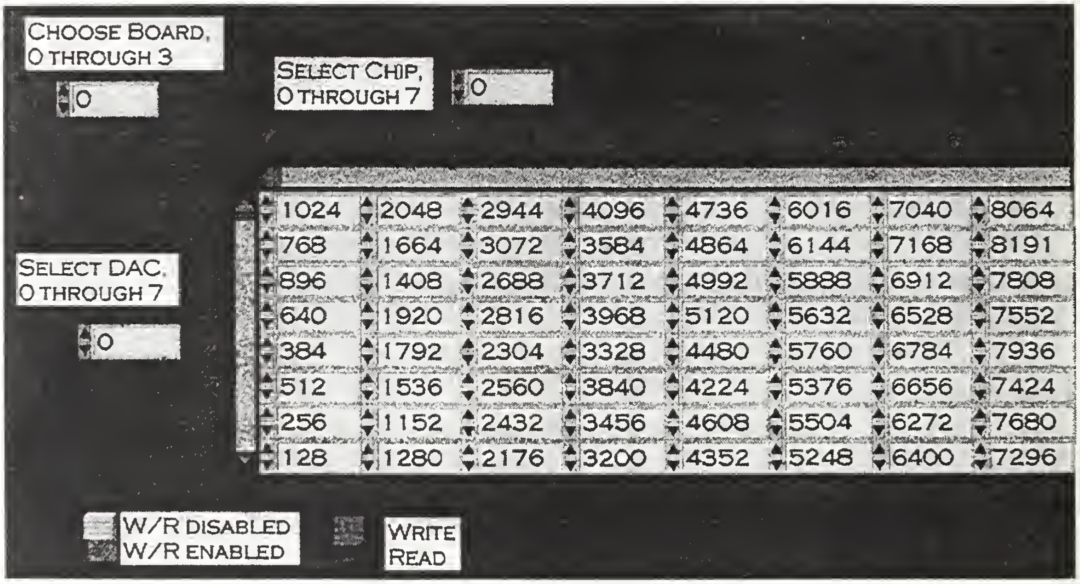


Figure 28 ODECL trigger output.

C. COMPARATOR BOARDS

To explore the outputs of the comparator boards a linear ramp voltage (+1 to +3 volts) was applied to the input of the boards and the outputs were latched using the ODECL trigger. The comparator boards were then programmed using the procedures described in Chapter III to provide a linear output. Figure 29 shows the LabVIEW vi Linear_4G front panel displaying the linear values for this test. The Choose Board switch selects the desired board to program. (Board 0 for mod65, board 1 for mod64, board 2 for mod 63 and board 3 for the parity comparator board.) The DAC and Chip select switches select the desired DAC and Chip, respectively, on the board. The W/R Disabled switch is preset to disabled to prevent inadvertent writes and the Write/Read switch must be selected to write to modify the file on the computer.



CHOOSE BOARD, 0 THROUGH 3: 0

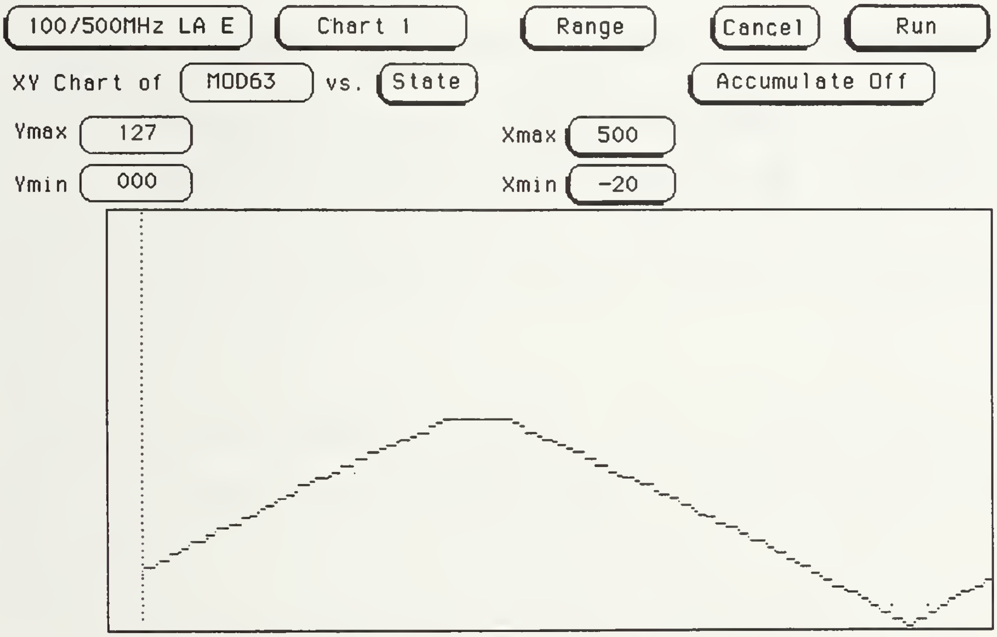
SELECT CHIP, 0 THROUGH 7: 0

SELECT DAC, 0 THROUGH 7: 0

1024	2048	2944	4096	4736	6016	7040	8064
768	1664	3072	3584	4864	6144	7168	8191
896	1408	2688	3712	4992	5888	6912	7808
640	1920	2816	3968	5120	5632	6528	7552
384	1792	2304	3328	4480	5760	6784	7936
512	1536	2560	3840	4224	5376	6656	7424
256	1152	2432	3456	4608	5504	6272	7680
128	1280	2176	3200	4352	5248	6400	7296

Figure 29 Linear_4G LabVIEW vi showing linear values.

The results from this test are shown in Figure 30. Note the nonlinearities at the lower right portion of the sweep. These are caused by comparator seven comparators being triggered instead of the desired number, six. Only one sweep of the voltage is shown but the nonlinearity repeats on all additional sweeps also. Figure 31, Figure 32 and Figure 33 show the output for the mod64, mod65 and parity comparator boards, respectively, for the linearity tests. Each of these figures shows nonlinearities over certain sections of the transfer curve displayed. The x-axis of the figures represents the sample number (the logic analyzer used to create these charts is synchronized to the digital antenna system), and the y-axis a digital representation of the number of comparators turned on. This corresponds to one of the folds shown in Figure 4.



XY Chart of MOD63 vs. State

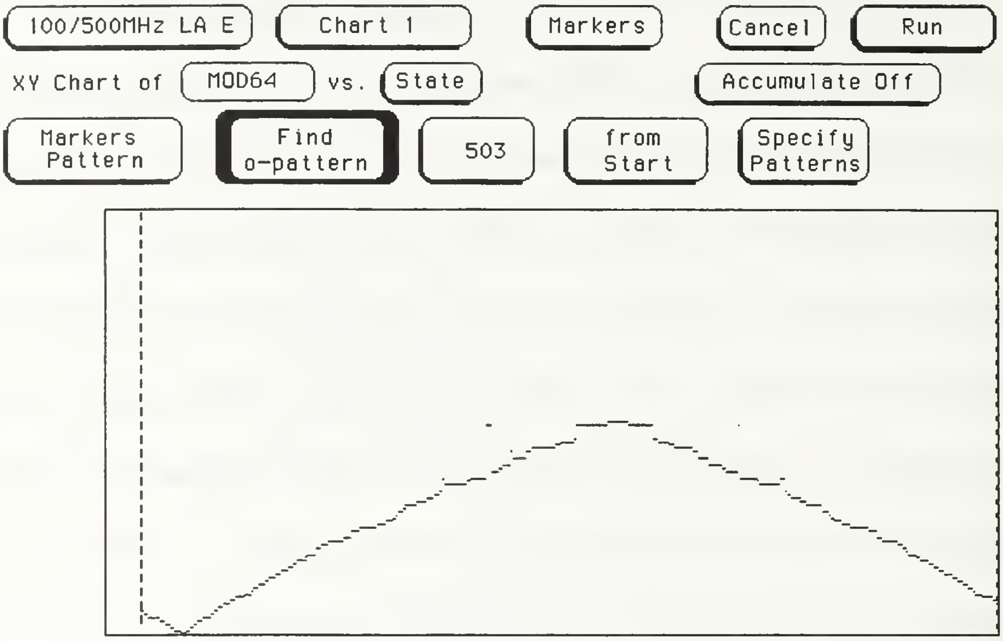
Ymax 127

Ymin 000

Xmax 500

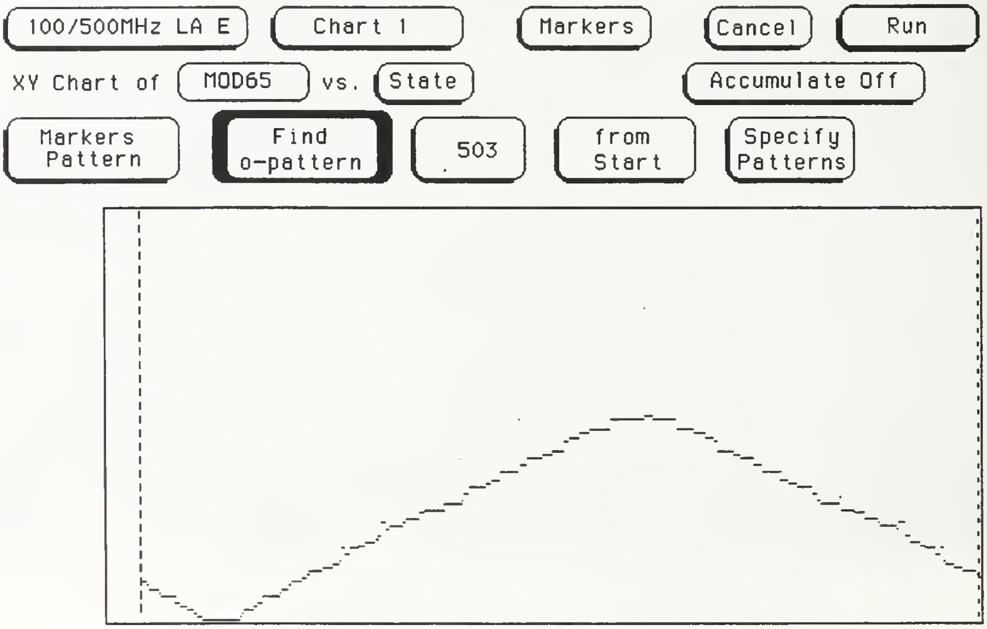
Xmin -20

Figure 30 Mod63 comparator board linear program test.



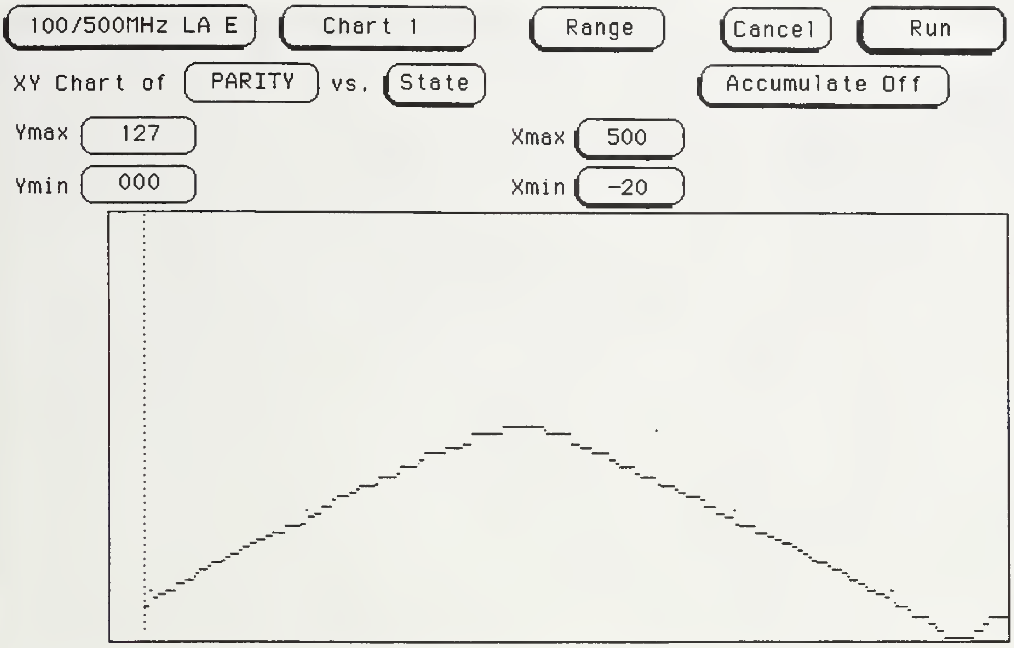
XY Chart of MOD64 vs. State

Figure 31 Mod64 comparator board linear program test.



XY Chart of MOD65 vs. State

Figure 32 Mod65 comparator board linear program test.



XY Chart of PARITY vs. State

Ymax 127

Ymin 000

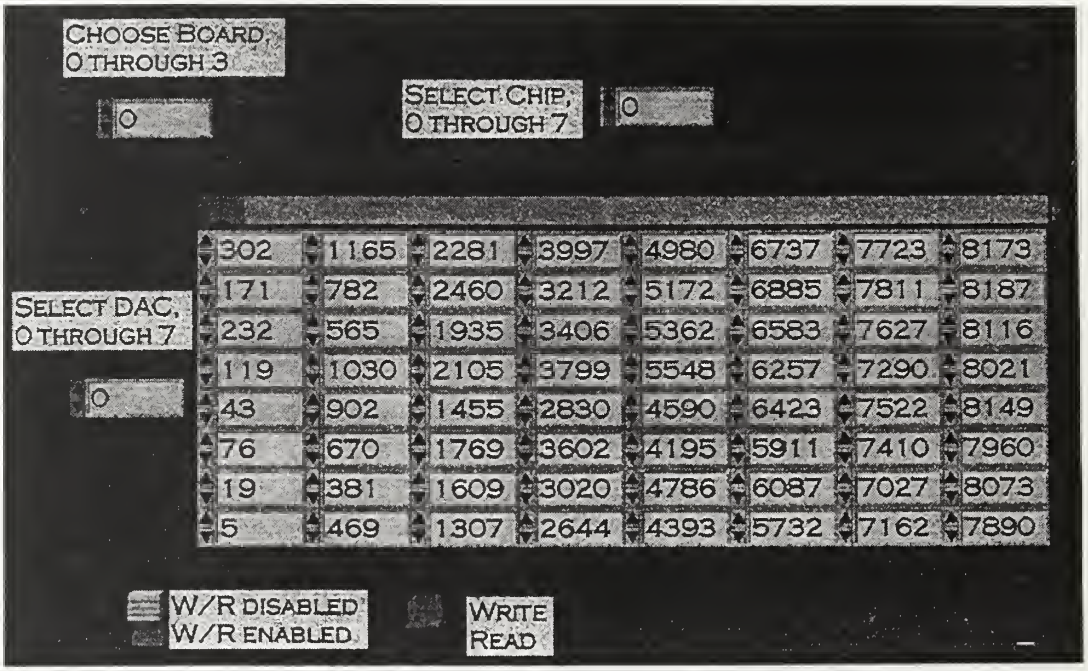
Xmax 500

Xmin -20

Figure 33 Parity comparator board linear program test.

The second step in verifying the comparator board outputs was to program the boards with the values computed to account for the sin^2 output of the MZIs. These values were computed using the MATLAB code included in Appendix A. Figure 34 shows the LabVIEW vi Control_4G control panel for the mod65 comparator board displaying the calculated comparator values.

Figure 35, Figure 36 and Figure 37 provide views of this same vi displaying the comparator level values for the mod64, mod63 and parity comparator boards respectively.



CHOOSE BOARD, 0 THROUGH 3

0

SELECT CHIP, 0 THROUGH 7

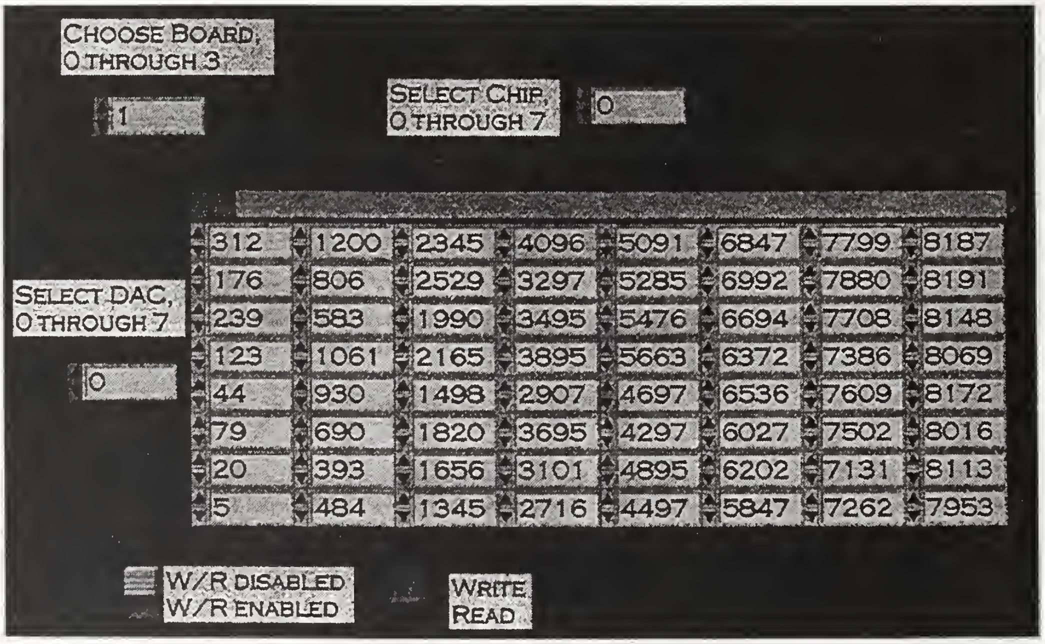
0

SELECT DAC, 0 THROUGH 7

0

302	1165	2281	3997	4980	6737	7723	8173
171	782	2460	3212	5172	6885	7811	8187
232	565	1935	3406	5362	6583	7627	8116
119	1030	2105	3799	5548	6257	7290	8021
43	902	1455	2830	4590	6423	7522	8149
76	670	1769	3602	4195	5911	7410	7960
19	381	1609	3020	4786	6087	7027	8073
5	469	1307	2644	4393	5732	7162	7890

Figure 34 LabVIEW Control_4G vi showing mod65 comparator levels.



CHOOSE BOARD, 0 THROUGH 3

1

SELECT CHIP, 0 THROUGH 7

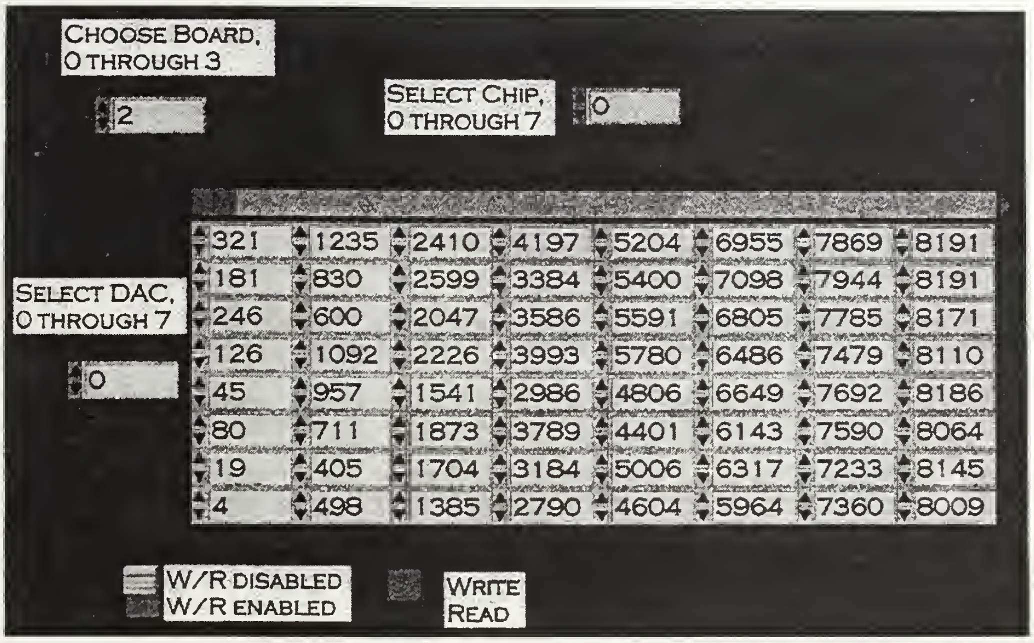
0

SELECT DAC, 0 THROUGH 7

0

312	1200	2345	4096	5091	6847	7799	8187
176	806	2529	3297	5285	6992	7880	8191
239	583	1990	3495	5476	6694	7708	8148
123	1061	2165	3895	5663	6372	7386	8069
44	930	1498	2907	4697	6536	7609	8172
79	690	1820	3695	4297	6027	7502	8016
20	393	1656	3101	4895	6202	7131	8113
5	484	1345	2716	4497	5847	7262	7953

Figure 35 LabVIEW Control_4G vi showing mod64 comparator levels.



CHOOSE BOARD, 0 THROUGH 3

2

SELECT CHIP, 0 THROUGH 7

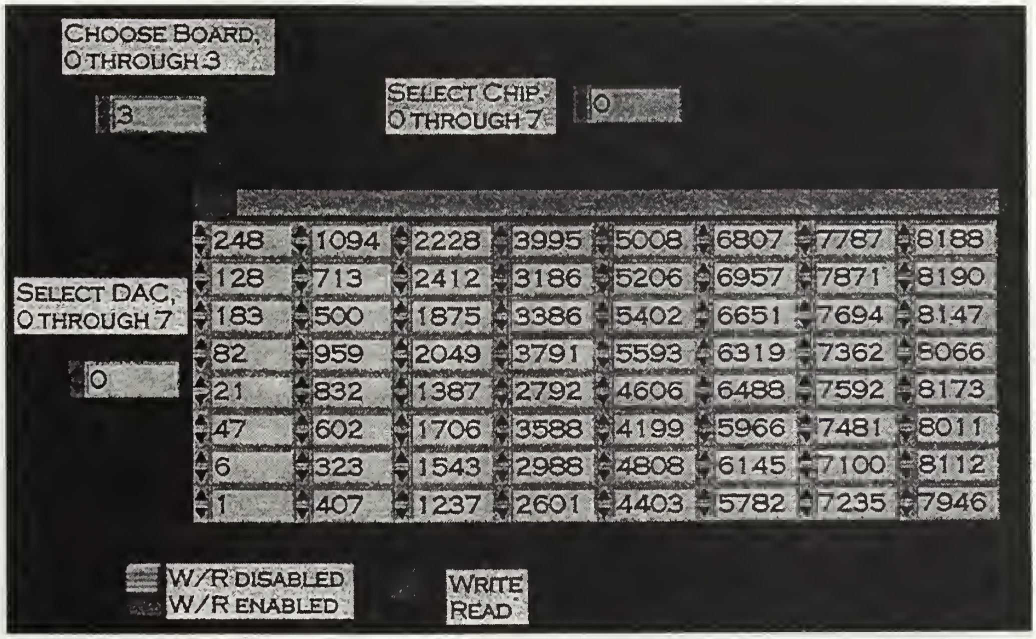
0

SELECT DAC, 0 THROUGH 7

0

321	1235	2410	4197	5204	6955	7869	8191
181	830	2599	3384	5400	7098	7944	8191
246	600	2047	3586	5591	6805	7785	8171
126	1092	2226	3993	5780	6486	7479	8110
45	957	1541	2986	4806	6649	7692	8186
80	711	1873	3789	4401	6143	7590	8064
19	405	1704	3184	5006	6317	7233	8145
4	498	1385	2790	4604	5964	7360	8009

Figure 36 LabVIEW Control_4G vi showing mod63 comparator levels.



CHOOSE BOARD, 0 THROUGH 3

3

SELECT CHIP, 0 THROUGH 7

0

SELECT DAC, 0 THROUGH 7

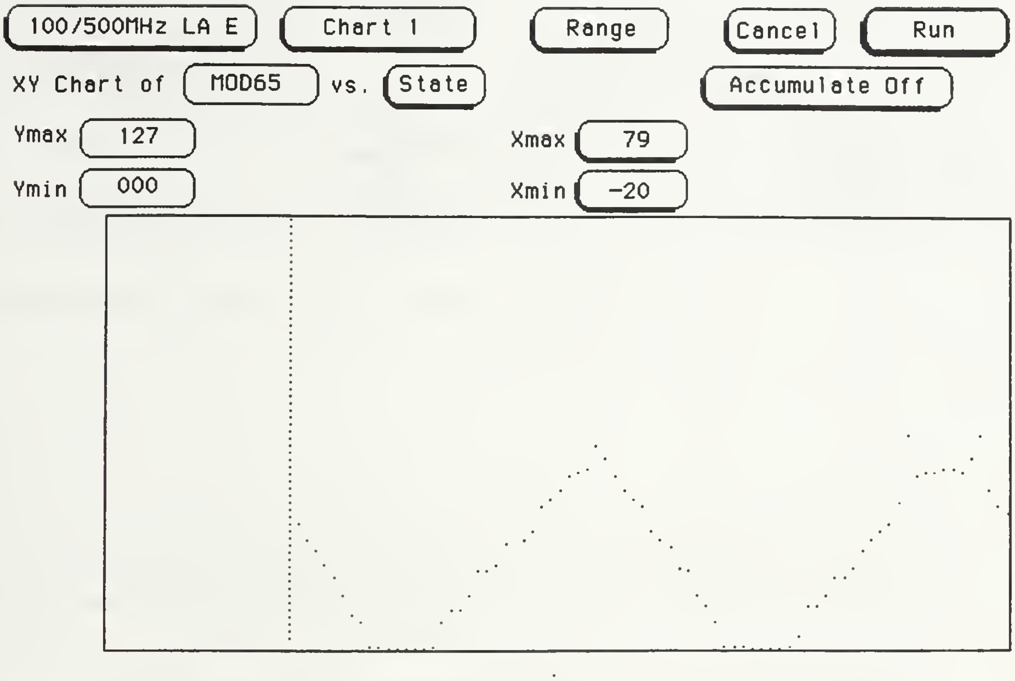
0

248	1094	2228	3995	5008	6807	7787	8188
128	713	2412	3186	5206	6957	7871	8190
183	500	1875	3386	5402	6651	7694	8147
82	959	2049	3791	5593	6319	7362	8066
21	832	1387	2792	4606	6488	7592	8173
47	602	1706	3588	4199	5966	7481	8011
6	323	1543	2988	4808	6145	7100	8112
1	407	1237	2601	4403	5782	7235	7946

Figure 37 LabVIEW Control_4G vi showing parity comparator levels.

The outputs corresponding the above input levels and the associated d.c. restoration circuitry inputs is shown in Figure 38 for the mod65, Figure 39 for the mod64, Figure 40 for the mod63 and Figure 41 for the parity comparator boards. The input to the MZIs was a 20 kHz triangular wave from 0 - 10 VDC. The noise effects of the optical subsystem are apparent in the nonlinearity of what should have been a binary representation of the OSNS residues created by the MZI folding circuit and linearized by the calculated comparator board levels. That is, given a $\sin^2$ input, the output should be a linear triangular wave.

The distortion evident on these waveforms is the result of the noise observed earlier on the pulsed waveforms arriving from the optical subsection.



XY Chart of MOD65 vs. State

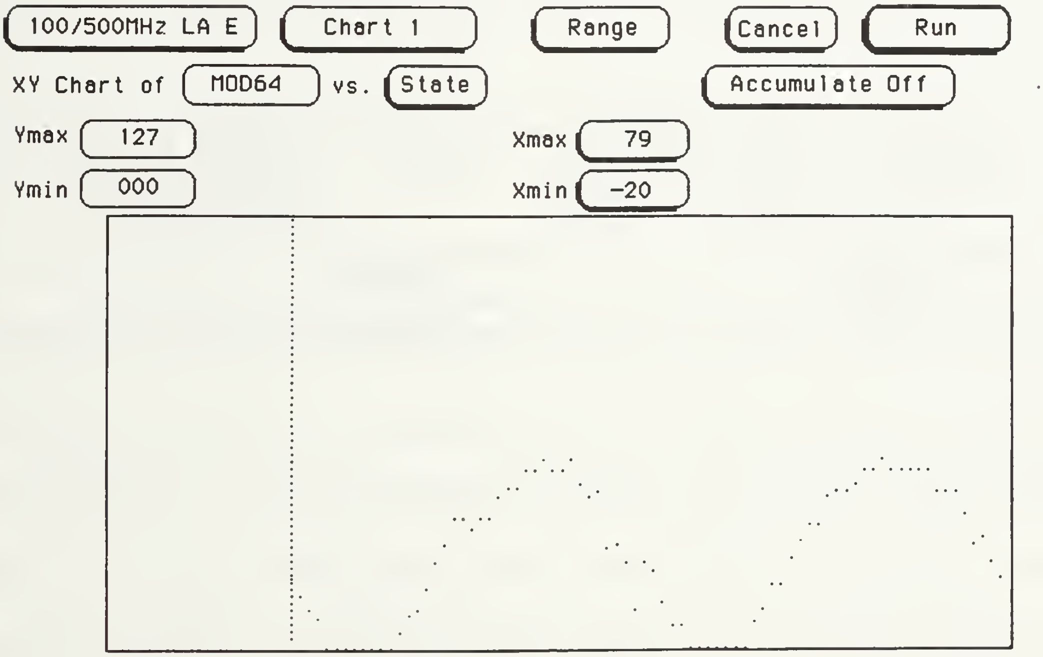
Ymax 127

Ymin 000

Xmax 79

Xmin -20

Figure 38 Mod65 comparator board output with triangle input to MZI.



XY Chart of MOD64 vs. State

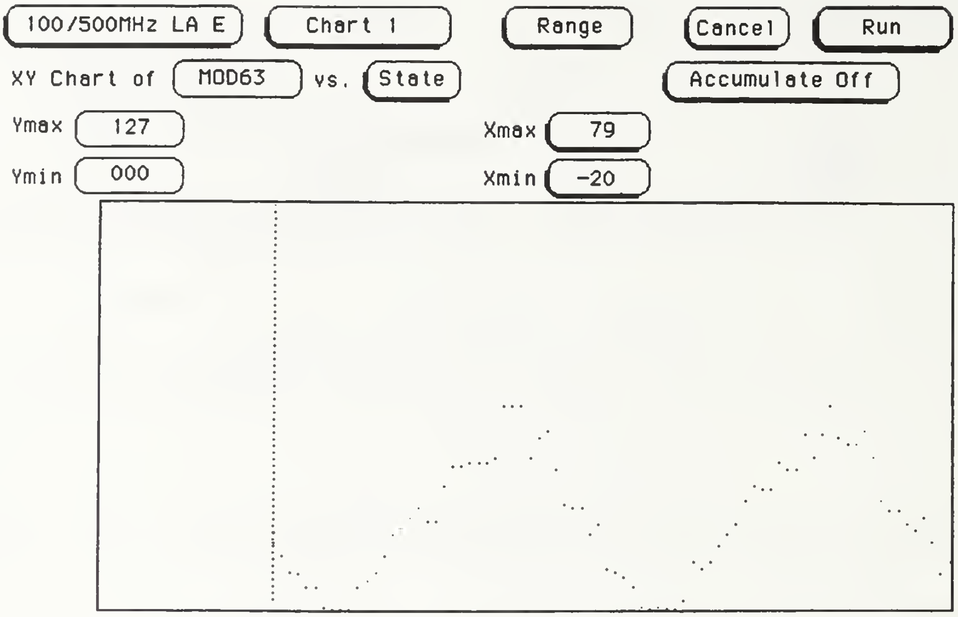
Ymax 127

Ymin 000

Xmax 79

Xmin -20

Figure 39 Mod64 comparator board output with triangle input to MZI.



XY Chart of MOD63 vs. State

Accumulate Off

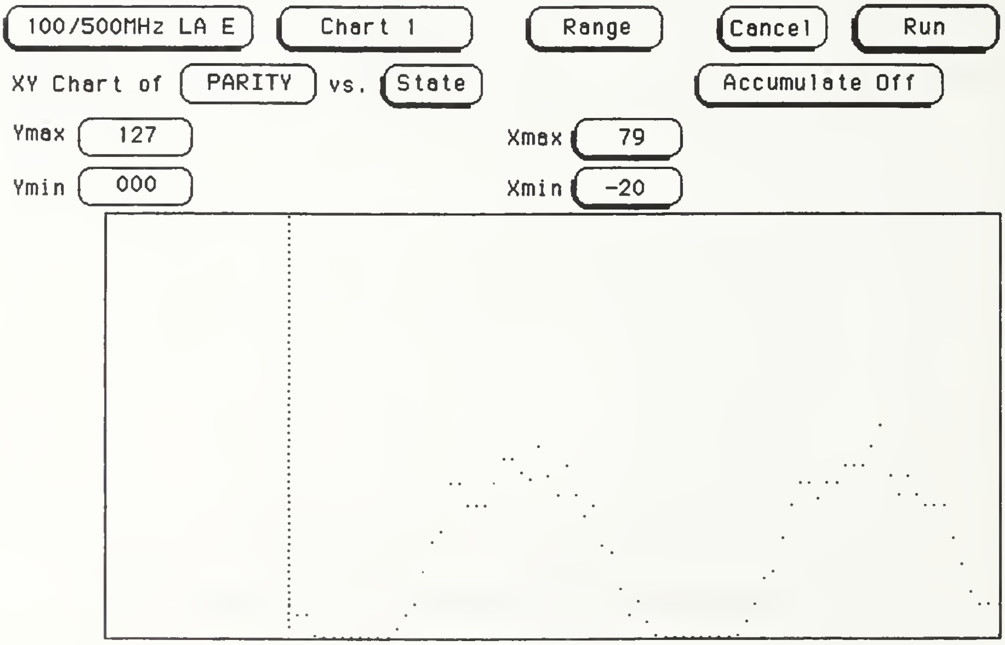
Ymax 127

Ymin 000

Xmax 79

Xmin -20

Figure 40 Mod63 comparator board output with triangle input to MZI.



XY Chart of PARITY vs. State

Accumulate Off

Ymax 127

Ymin 000

Xmax 79

Xmin -20

Figure 41 Parity comparator board output with triangle input to MZI.

D. OSNS FPGA

Foundation ActiveCAD, the FPGA design tool created by ALDEC, allows the user to test the design prior to programming the target FPGA. The result of the functional test of the OSNS-to-binary FPGA is shown in Figure 42.

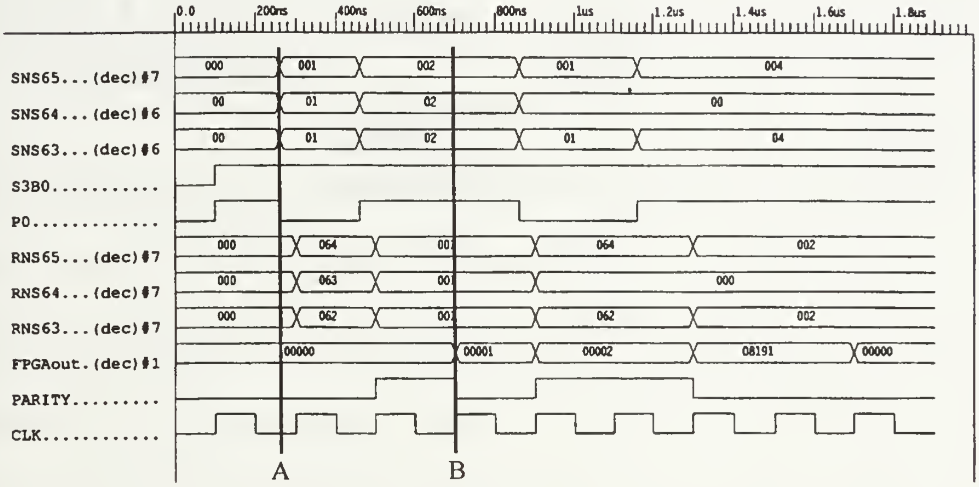
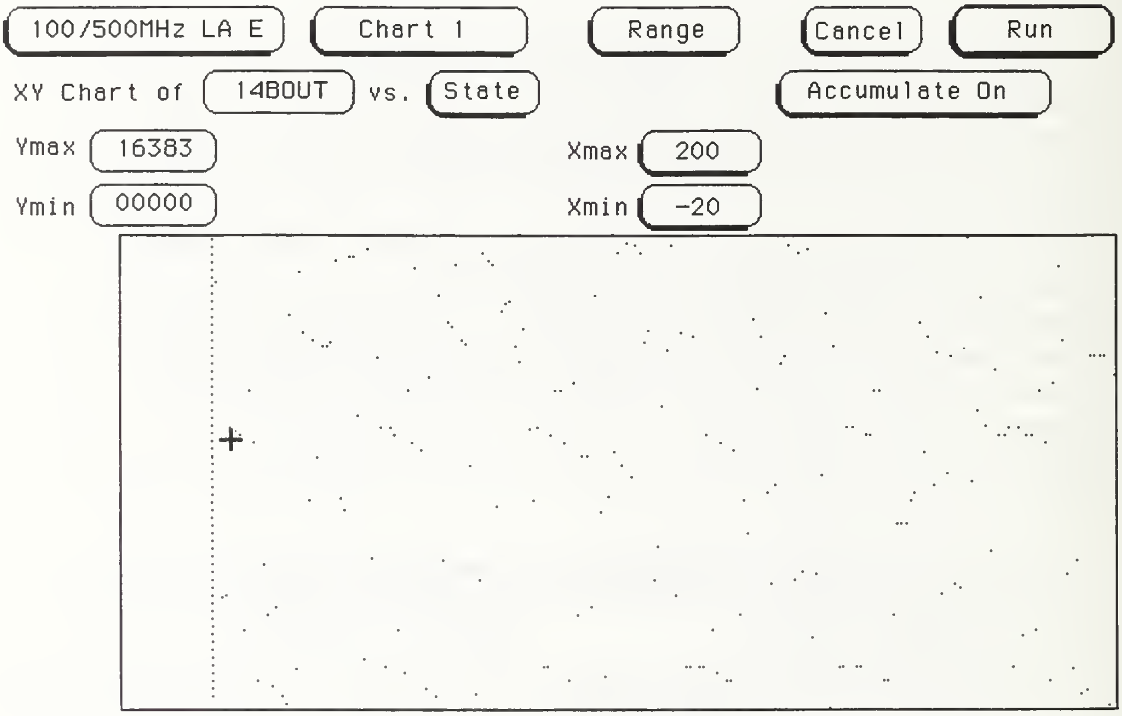


Figure 42 OSNS FPGA design functional test output.

Note that three clock cycles (bottom trace) after the inputs, labeled SNS65, SNS64, SNS63 and S3B0 (parity comparator board LSB), change to 001, 01, 01 (labeled A on Figure 42) and High (1), respectively, the 14-bit output (FPGAout) changes to 00001 to reflect this. At this time (labeled B on Figure 42) the parity output (PARITY) changes to a Low (0) to reflect the even parity of the SNS63 and S3B0. Also note the parity output changing to a High when the input of 002, 02, 02 and High are reflected in the output of 00002. When the input is 004, 04, 04 and High (corresponding to an input value for the OSNS system of 2^{14}) FPGAout goes to zero and the parity output goes Low indicating a value greater than 2^{14}-1 which is the maximum allowable value in a 14-bit system.

The output of the OSNS FPGA with the input set up as described for the comparator board output is shown in Figure 43.



XY Chart of 14BOUT vs. State

Accumulate On

Ymax 16383

Ymin 00000

Xmax 200

Xmin -20

Figure 43 14-bit OSNS output with 20KHz triangle input to MZI.

The Xmax of this chart has been extended out to 200 samples in the effort to discern any underlying pattern. Readily apparent in this figure is the fact there is no correlation between the input 20 kHz triangle waveform and the final 14-bit digital output. Possible reasons for this lack of correlation will be discussed in the following conclusions chapter.

VII. CONCLUSIONS AND FOLLOW-ON WORK

A. CONCLUSIONS

A 14-bit OSNS-based Direct Digital Antenna is an ambitious project. All aspects of the design have to be exact to a tolerance of 1 in 2^{14}. The OSNS encoding/decoding algorithm is not forgiving of one-bit errors. For instance, with OSNS residues mod-65 = 1, mod-64 = 1 and mod-63 =1, the output will be 1. If, instead, the residues are mod-65 = 1, mod-64=0 and mod-63 = 1, the resultant output will be 8191. That is, a 1-bit error in the input residues results in a value difference equal to ½ the range of the system. This means that the noise on the data pulses arriving at the comparator boards from the d.c. restoration circuitry must be below one half of one LSB (5 mV). As shown in Chapter VI the incoming noise is at least ten times that amount. This is a major problem that needs to be rectified before work can continue.

The parity processing scheme used in this project only bases its decision on the least significant bits of the output of the mod-63 and parity comparator boards. This assumes that the comparator values from those two boards are within one level of each other. Table 4 contains a sample state output from the comparator boards on the test used in Chapter VI to test the project operation. This table shows that the mod-63 and parity outputs are in fact seldom within one level difference. This completely nullifies the parity scheme and does not allow for determination of sampling within decimation bands.

Label >	MOD65	MOD64	MOD63	PARITY
Base >	Decim	Decim	Decim	Decima
0	043	021	021	010
1	037	016	017	007
2	032	012	012	007
3	029	009	011	001
4	025	000	007	000
5	021	000	007	000
6	016	000	001	000
7	010	000	000	000
8	008	000	000	000
9	001	000	000	000
10	001	000	007	000
11	000	000	009	000
12	000	005	012	003
13	000	010	017	007
14	000	012	024	010
15	000	018	026	020
16	001	026	029	029
17	008	031	032	032
18	012	039	028	047
19	012	039	028	047
20	016	036	039	040
21	023	039	045	040
22	023	039	045	040
23	025	045	046	047
24	031	048	046	054
25	127	048	046	054
26	032	053	048	050
27	035	053	064	048
28	042	056	064	058
29	044	053	064	049
30	047	053	048	043
31	051	056	054	052
32	052	049	056	043
33	053	045	044	037
34	060	047	033	040
35	056	030	032	028
36	051	031	032	026
37	047	026	024	015
38	044	012	027	007
39	042	026	013	011
40	035	023	012	005
41	032	014	010	001
42	030	007	007	000
43	024	007	001	000
44	023	000	000	000
45	016	000	001	000

Table 4 Comparator board outputs.

As also shown in Chapter VI the outputs of the comparator boards are not linear as needed for the decoding algorithm to work. While this noise has less of a magnitude than the noise introduced from the optical subsection, it contributes to distortion of the output value. Noise problems in the optical subsection are discussed in Reference 3.

B. FOLLOW-ON WORK

The OSNS Direct Digital Antenna is a viable project. All simulations prove the theory behind the design to be sound. Refinements need to be made to reduce the noise levels in the optical subsection of the project. Recommendations include:

- Using a mode-locked laser to stimulate the Mach-Zehnder Interferometers.

- Using polarization preserving fiber-optic cable or polarization rotators on the existing fiber throughout the optical subsection up to the outputs of the circulators.

- Modifying the optical path so the optical circulators can be removed from the data paths.

- Modifying the OSNS decode encode/algorithm such that one-bit errors only produce a small change in the output (a Gray code approach).

- Redefining the OSNS decoding algorithm such that a greater variety of piecewise-prime sequences can be used. At this time the algorithm can only support sequences of 2^n-1, 2^2, 2^n+1 where n is an integer value.

- Properly triggering the parity interpolation scheme needs to be triggered properly to allow the FPGA data to settle prior to latching the parity input.

APPENDIX A. MATLAB PROGRAM

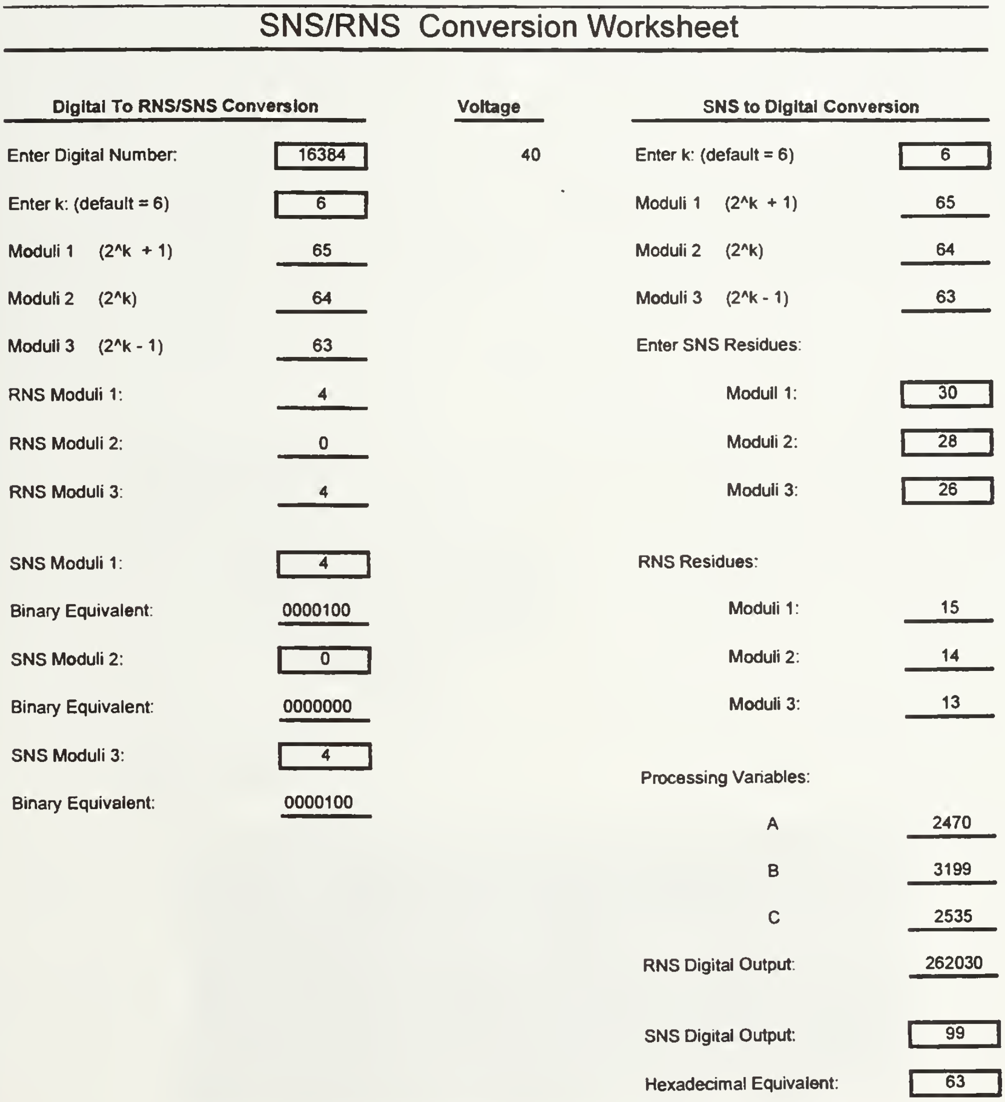
The following MATLAB program calculates the values for programming the comparator boards.

```
% Thesis program (complvls.m)
% This m-file generates the threshold values for the comparator levels
% of the comparator board.  The levels for the parity and mod 2^k -1
% channels are generated based on the percentage of LSB decimation
% width required.  The k and the desired percentage of decimation are
% required inputs.
% Program written by Bill Ringer

%Clear the processor
clear

% Get the required information
k = input('Enter the value for k: ')
n = input('Enter the desired percent of the decimation width: ')

% Determine the moduli numbers
m1 = 2^k -1;
m2 = 2^k;
m3 = 2^k +1;

% Create some processing matrices
Drange1 = m1*(0:1/m1:1);
Vin1 =pi*Drange1/(2*m1);
Drange2 = m2*(0:1/m2:1);
Vin2 =pi*Drange2/(2*m2);
Drange3 = m3*(0:1/m3:1);
Vin3 =pi*Drange3/(2*m3);

% Calculate the thresholds for 2^k and 2^k + 1
Vth2 = sin(pi*Drange2/(2*m2)).^2;
Vth3 = sin(pi*Drange3/(2*m3)).^2;

%Convert to a voltage
vlt64 = Vth2*2+1;
vlt65 = Vth3*2+1;
% Convert to a comparator level
```

```
mod64 = floor((vlt64 - 1) * 4096);
mod65 = floor((vlt65 -1) * 4096);

% Perform the calculations for the parity decimation bands
% First, get threshold value of mod.m1
Vth1 = sin(pi*Drange1/(2*m1)).^2;

% Then establish the difference based on the percentage of LSB
% decimation
dVin = Vin1(2) - Vin1(1);
dtmod= n * dVin / 200;

% Establish some temporary processing variables t1 and t2
t1= Vin1 - dtmod;
t2= Vin1 + dtmod;

% Create the vector
tmod=[t1;t2];
tmod=reshape(tmod,1,2*length(Vin1));
tmod=tmod(:,2:length(tmod)-1);
PVth=sin(tmod).^2;

% Establish the separate vectors
tm63= [2:2:124];
tpar = [1:2:125 126];

% Convert to a voltage
vlt63 = PVth(tm63)*2+1;
vltpar = PVth(tpar)*2+1;

% Convert to a comparator level
mod63 = floor((vlt63 - 1) * 4096);
parity = ceil((vltpar -1) * 4096);

% Override upper and lower parity levels to create larger bands at the
% curves
parity(64)=8189;
parity(1)=0002;

%Pad the mod 63 to get 64 outputs to program the comparator board
mod63 = [mod63 8191 8191];

% Pad the mod64 to get 64 outputs to program the comparator board
mod64 = [mod64 8191];
```

APPENDIX B. OSNS MICROSOFT EXCEL RESULTS

The following shows typical results from an Excel worksheet used to compute the residues in a 2^k based OSNS system or calculate the output based on known residues.



SNS/RNS Conversion Worksheet

Digital To RNS/SNS Conversion

Enter Digital Number:	16384
Enter k: (default = 6)	6
Moduli 1 (2^k + 1)	65
Moduli 2 (2^k)	64
Moduli 3 (2^k - 1)	63
RNS Moduli 1:	4
RNS Moduli 2:	0
RNS Moduli 3:	4
SNS Moduli 1:	4
Binary Equivalent:	0000100
SNS Moduli 2:	0
Binary Equivalent:	0000000
SNS Moduli 3:	4
Binary Equivalent:	0000100

Voltage

40

SNS to Digital Conversion

Enter k: (default = 6)	6
Moduli 1 (2^k + 1)	65
Moduli 2 (2^k)	64
Moduli 3 (2^k - 1)	63
Enter SNS Residues:	
Moduli 1:	30
Moduli 2:	28
Moduli 3:	26
RNS Residues:	
Moduli 1:	15
Moduli 2:	14
Moduli 3:	13
Processing Variables:	
A	2470
B	3199
C	2535
RNS Digital Output:	262030
SNS Digital Output:	99
Hexadecimal Equivalent:	63

APPENDEX C. LABVIEW VIRTUAL INSTRUMENTS

A. *ADC_EXE.VI*

The *ADC_exe* is used to send the bit pattern generated by *SNS_control3f* and *SNS_control4g* vi's along with the associated task ID to the *DIO Port_Write* vi. The *Port_write* vi is an LabVIEW provided vi which sends the data to the correct pins on the DIO-96 interface board. Figure 44 shows the front panel and the connector plane interface for the *ADC_exe* vi and Figure 45 shows the internal configuration diagram. The inputs to the vi are task ID and Pattern and this information is routed to the *DIO Port_Write* vi.

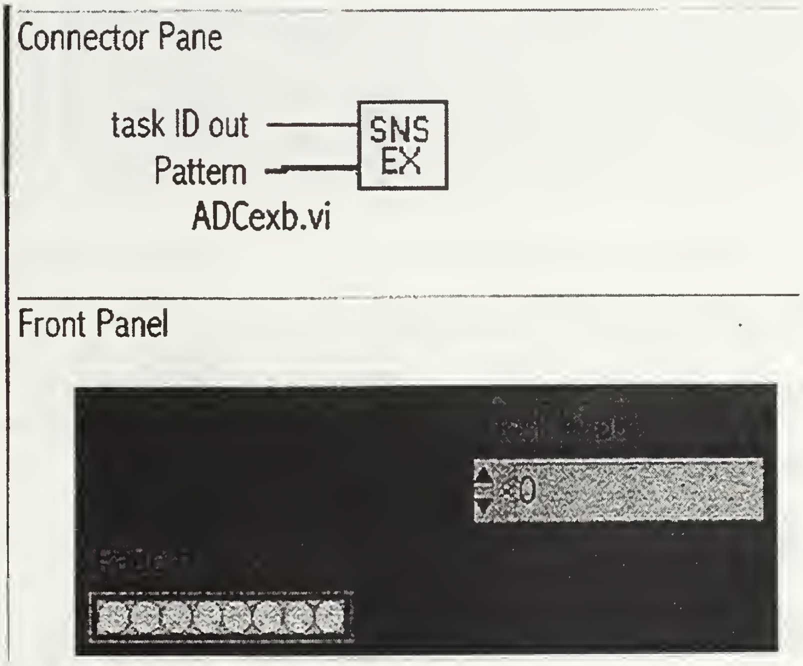


Figure 44 ***ADC_exe*** **vi interface and front panel.**

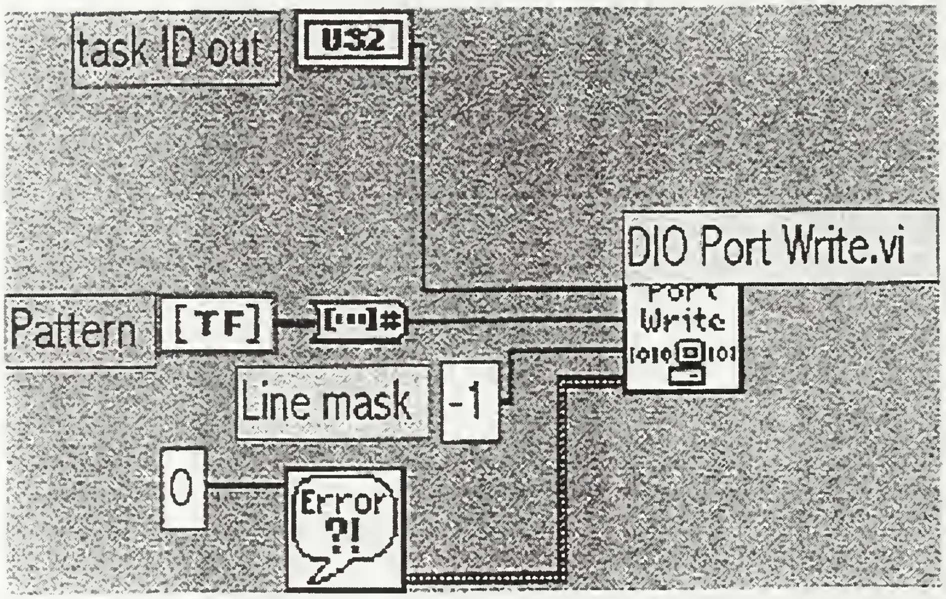


Figure 45 ***ADC_exe*** **vi internal configuration diagram.**

B. *SNS_CONTROL3F* VI

This vi takes the various settings of values (described below) and builds one binary array from the data. This array, or pattern, is then routed to the *ADC_exe* vi described above. The control panel for the *SNS_Control3f* vi is shown in Figure 46.

Figure 47 displays the connector plane interface. The inputs are received from the *SNS_Control4g* and *SNS_Runf* vi's and include the following

- Load Enable - A single T/F value to enable the desired comparator board.

- Data Level - This 13-bit binary representation of the desired comparator level.

- Select Chip - 3-bit binary number representing the desired chip.

- DAC Select - 3-bit binary number representing the desired DAC.

- Load Enable - 1-bit T/F value used to load the MAX-547 DAC buffers.

- Write - 1-bit T/F value used to write to the MAX-547 input buffers.

- Task ID - Identification number generated by LabVIEW for selecting the correct ports on the DIO-96 interface card.

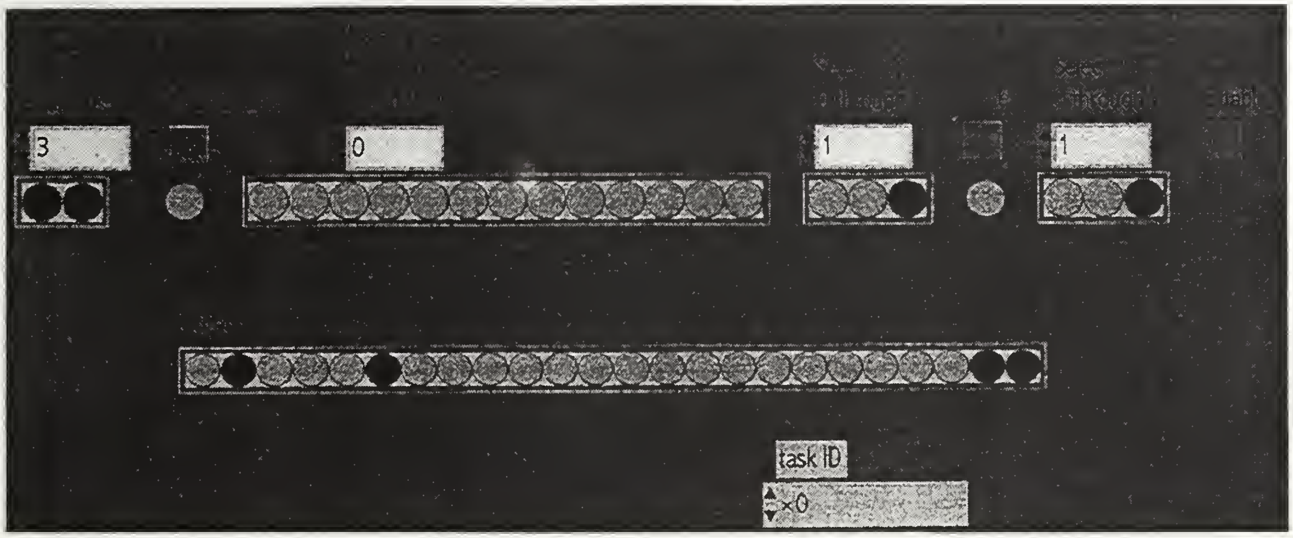


Figure 46 ***SNS_Control*** **vi front panel.**

The internal configuration routing diagram for the *SNS_Control3f* vi is shown in Figure 48. The values representing each of the inputs to the Build Array tool in the upper right hand section of the figure are all represented on the control panel so the process can be monitored. The Build Array tool creates the bit pattern from the input data and routs it and the task ID number to the *SNS_exe* vi.

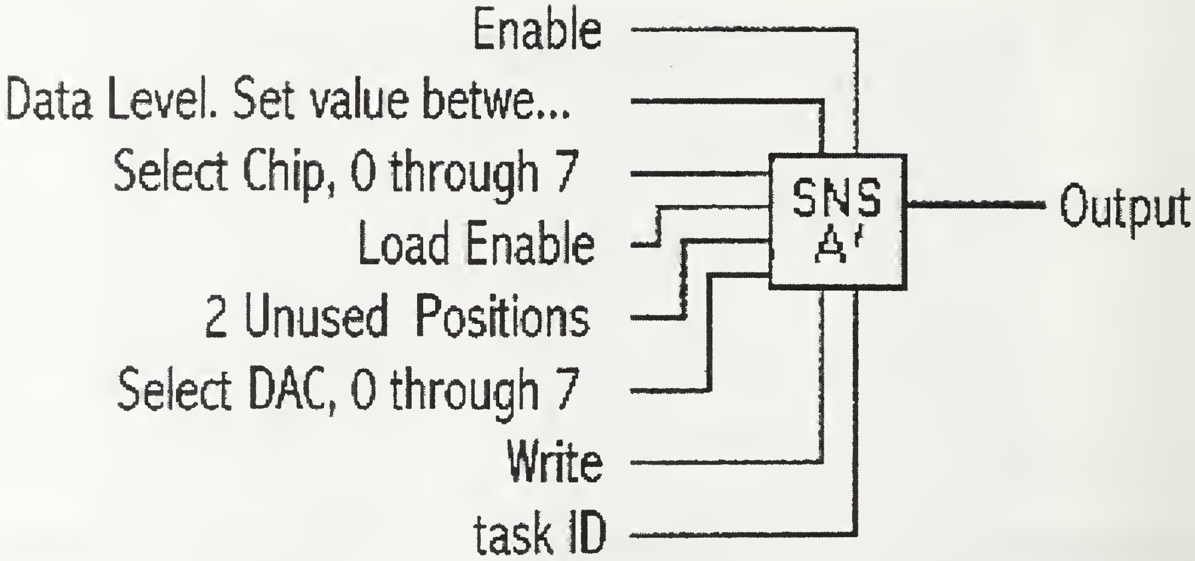


Figure 47 *SNS_Control3f* **vi connector plane interconnect.**

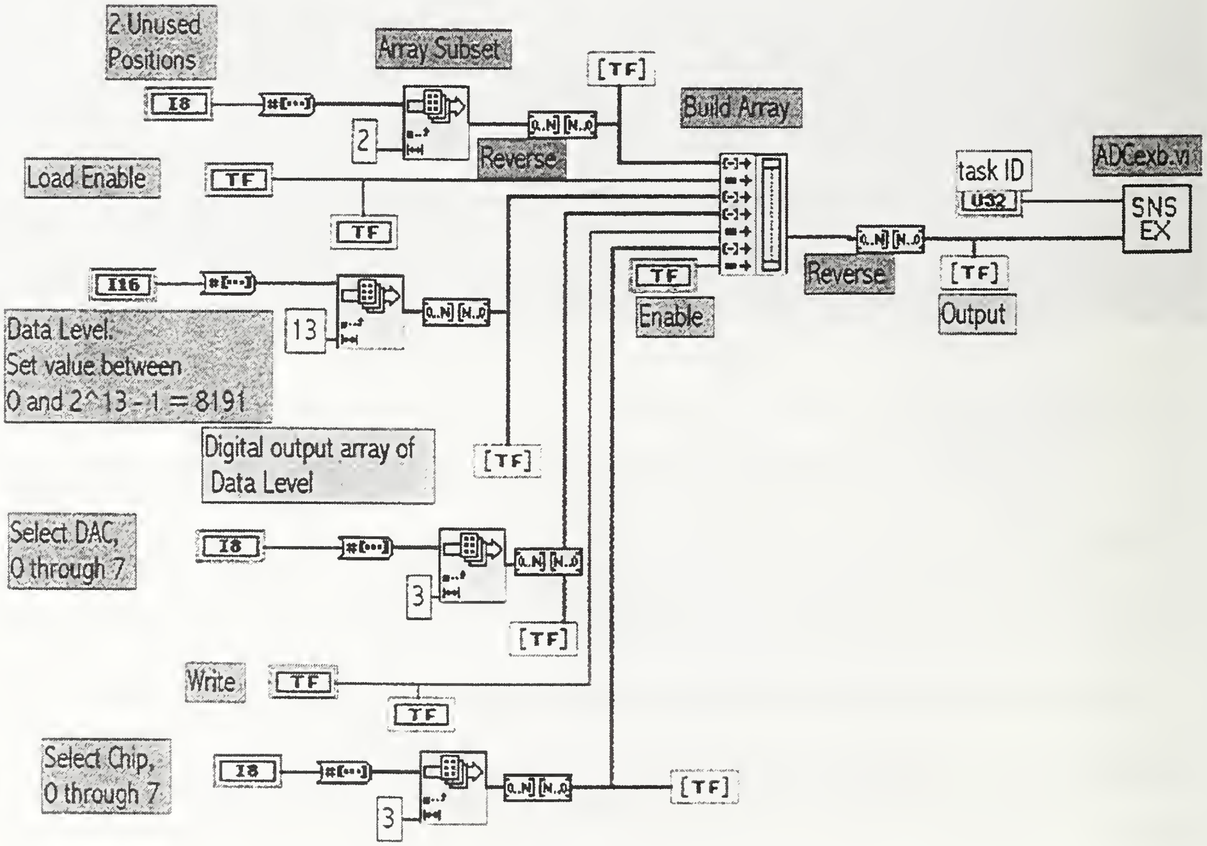
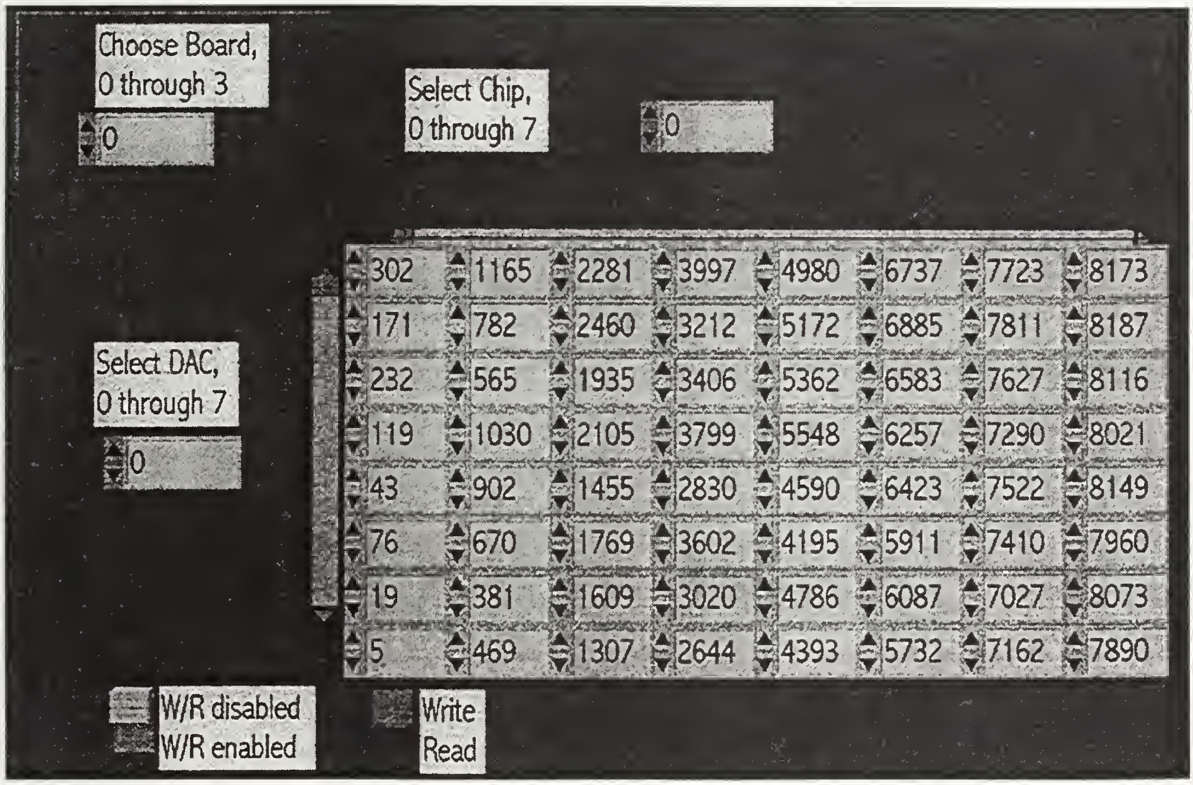


Figure 48 *SNS_Control3f* **vi internal routing diagram.**

C. *SNS-CONTROL4G* VI

This vi allows the operator to set the levels of the individual comparators on the selected comparator board. This vi does not write directly to the board, instead it writes to a file sns0,1,2,3.wri which is then read by the *SNS_Runf* vi and used to program the board. Figure 49 shows the front panel for the *SNS_Control4g* vi.



Choose Board, 0 through 3: 0

Select Chip, 0 through 7: 0

Select DAC, 0 through 7: 0

302	1165	2281	3997	4980	6737	7723	8173
171	782	2460	3212	5172	6885	7811	8187
232	565	1935	3406	5362	6583	7627	8116
119	1030	2105	3799	5548	6257	7290	8021
43	902	1455	2830	4590	6423	7522	8149
76	670	1769	3602	4195	5911	7410	7960
19	381	1609	3020	4786	6087	7027	8073
5	469	1307	2644	4393	5732	7162	7890

Figure 49 ***SNS_Control4g*** **vi front panel.**

The operator can select the desired comparator board using the Choose Board selector. The Select DAC and Select Chip selectors are used for modifying individual comparator level values. To read the sns0,1,2,3.wri from the disk, the Write/Read button must be set to Read (default value), the WR disabled/WR enabled must be set to WR enabled (not default value) and the vi run. For writing information to the sns0,1,2,3.wri files the Write/Read button must be set to Write and the WR enabled/WR disabled set to WR enabled.

Figures 50 - 56 show the internal configuration diagrams for the *SNS_Control4g* vi.

The large outer structure in Figure 50 is called a frame. This is the outer frame. When the vi is run, the frame executes serially starting at frame 0 through the rest of the frames with the internal operations in each frame completing prior to the next frame executing.

Figure 50 shows frame 1 and Figure 51 shows frame 0. The structure in the upper right of the outer frame is a selector box. The actions performed inside the box are dependent on the input variable, in this case WR disabled/WR enabled. Figure 50 shows that, with the WR disabled set to True, there is no operation conducted inside the box. The operations that occur when the WR disable is set to False are shown in Figure 56.

In the lower right section of Figure 50 is located the inner frame. This frame structure is responsible for writing to the DIO-96 interface board and, because the task ID input is disconnected inside the frame, is not functional in this vi. The operation of these frames will be amplified further in the discussion of the *SNS_Runf* vi.

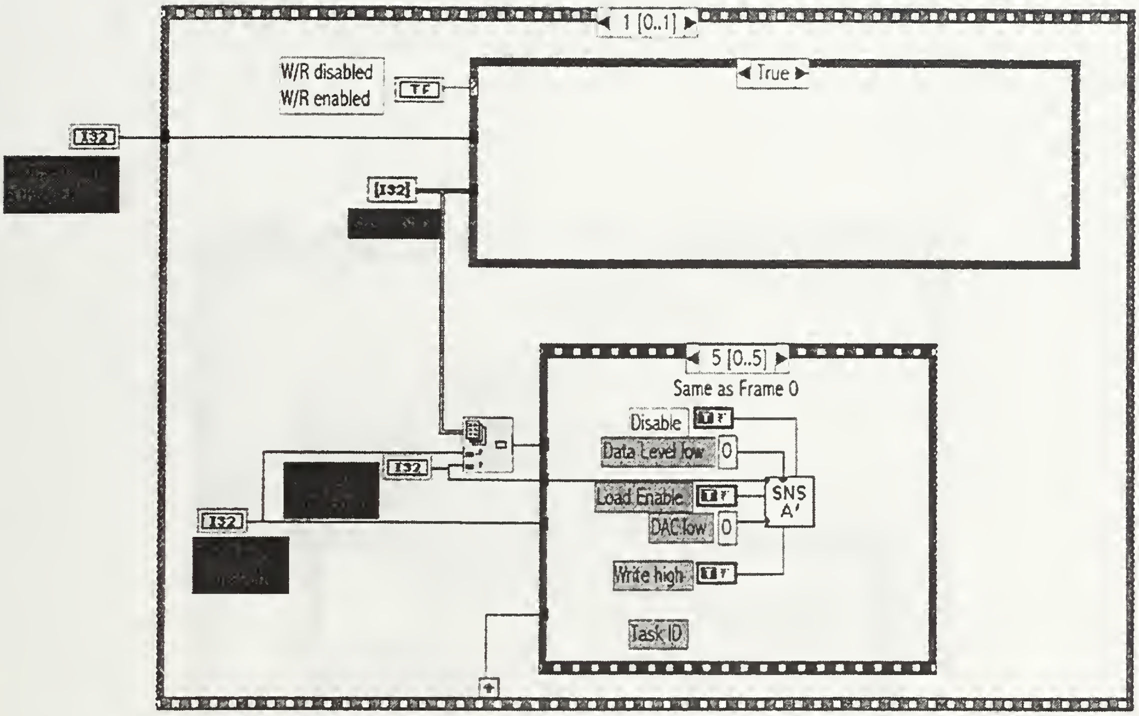


Figure 50 *SNS_Control4g* vi internal diagram (1 of 7).

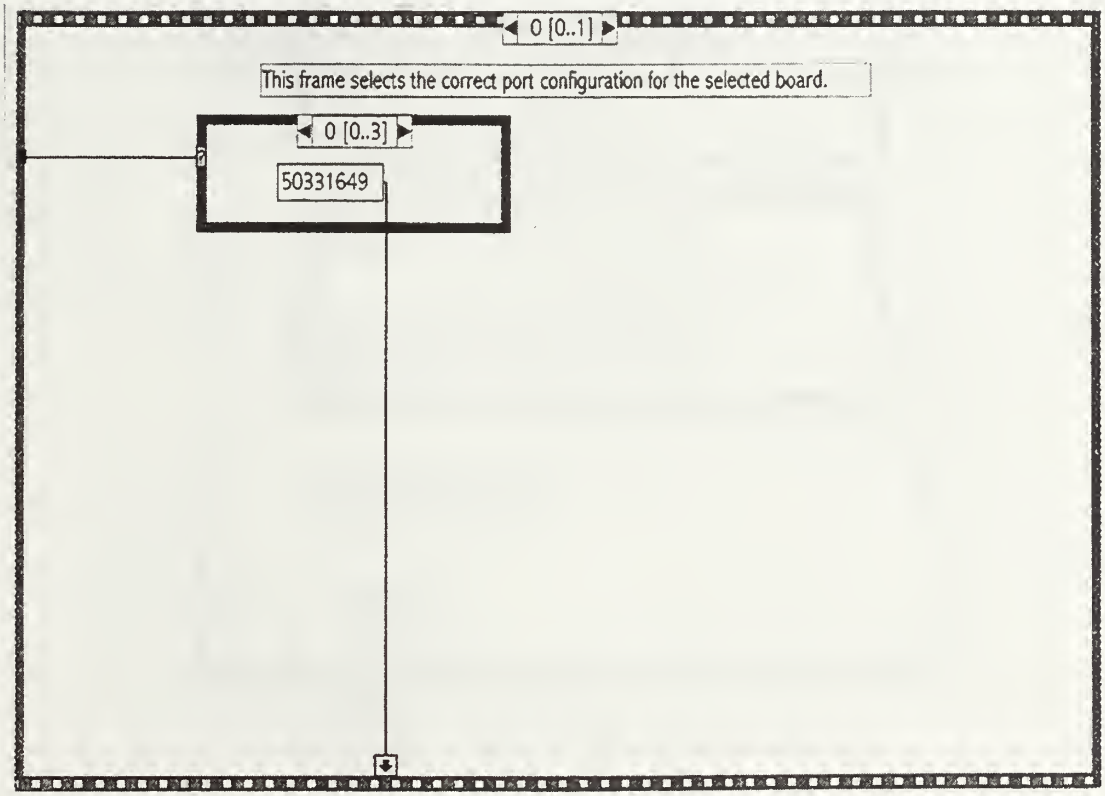

0 [0..1]
This frame selects the correct port configuration for the selected board.
0 [0..3]
50331649

Figure 51 *SNS_Control4g* vi internal diagram (2 of 7).

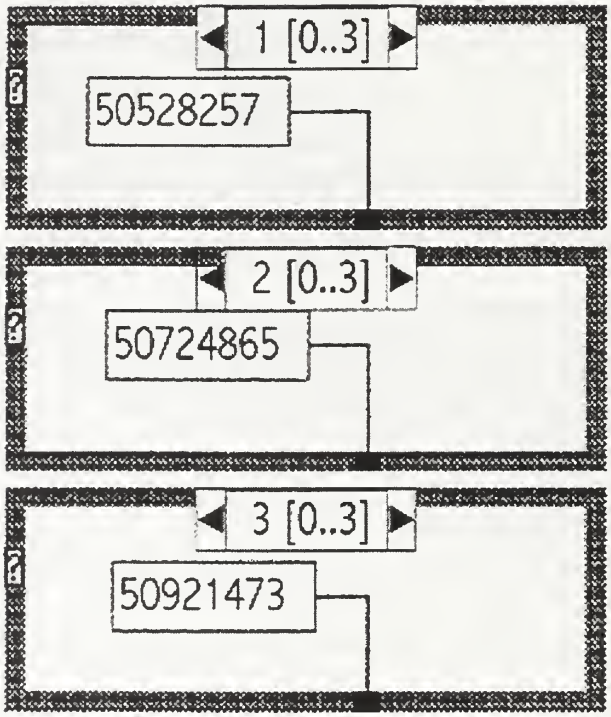


Figure 52 *SNS_Control4g* vi internal diagram (3 of 7).

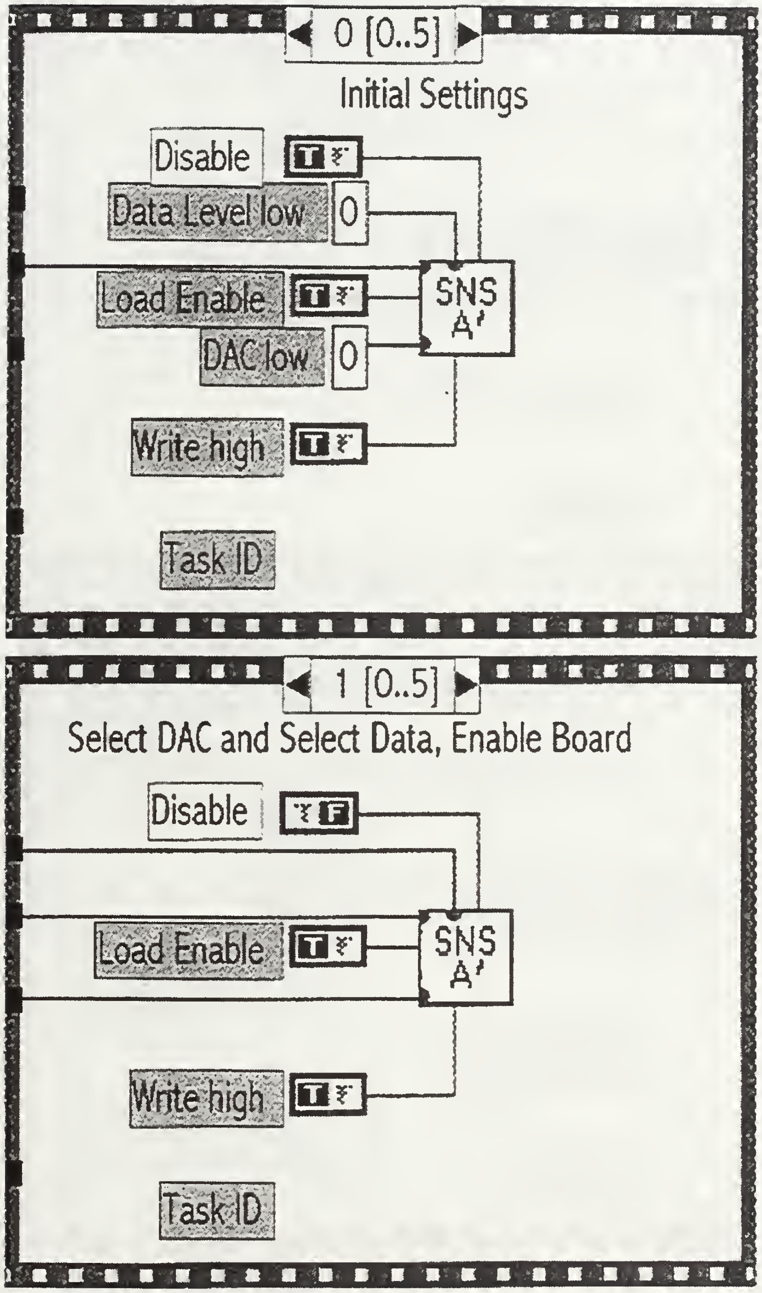


Figure 53 ***SNS_Control4g*** **vi internal diagram (4 of 7).**

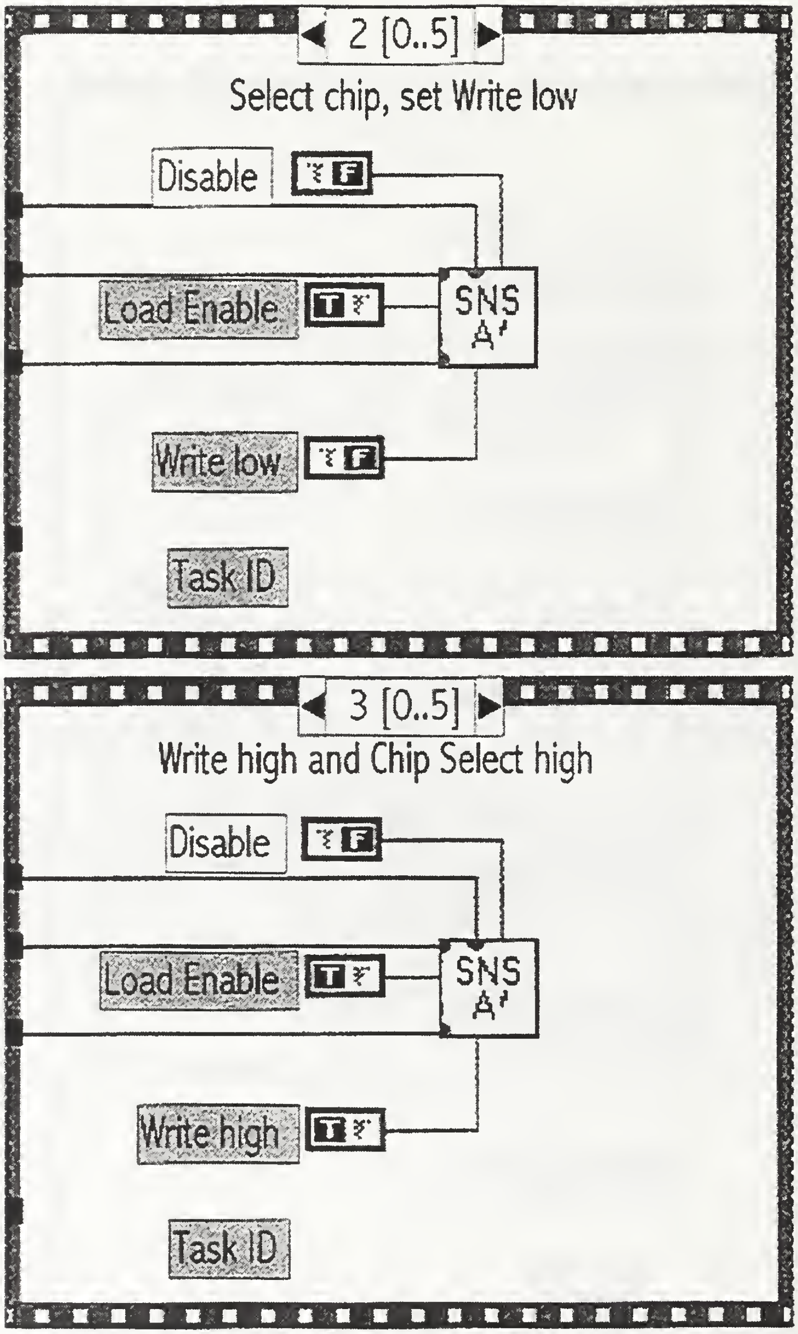


Figure 54 *SNS_Control4g* **vi internal diagram (5 of 7).**

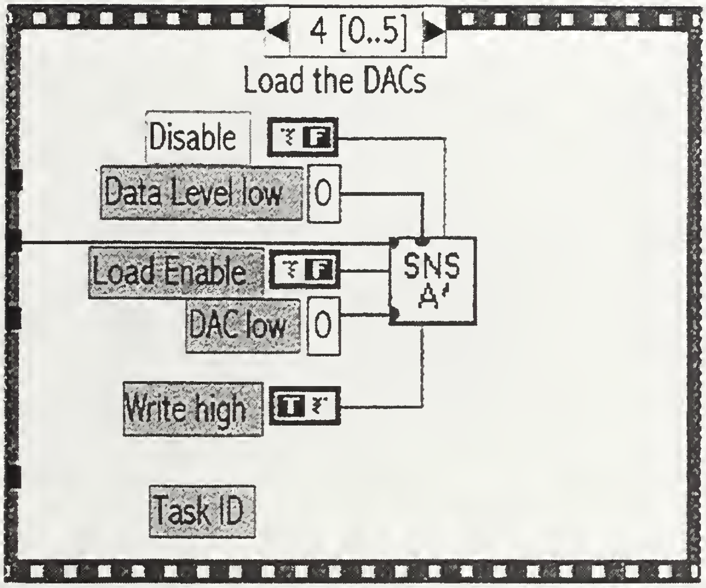


Figure 55   *SNS_Control4g* vi internal diagram (6 of 7).

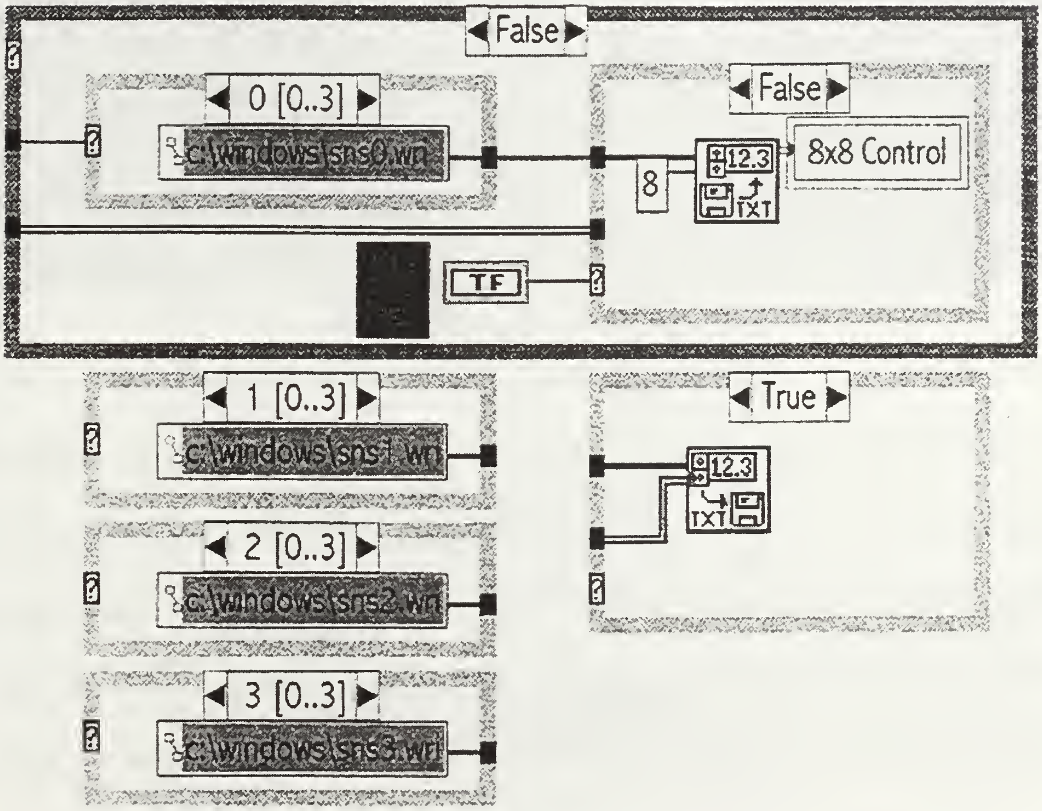
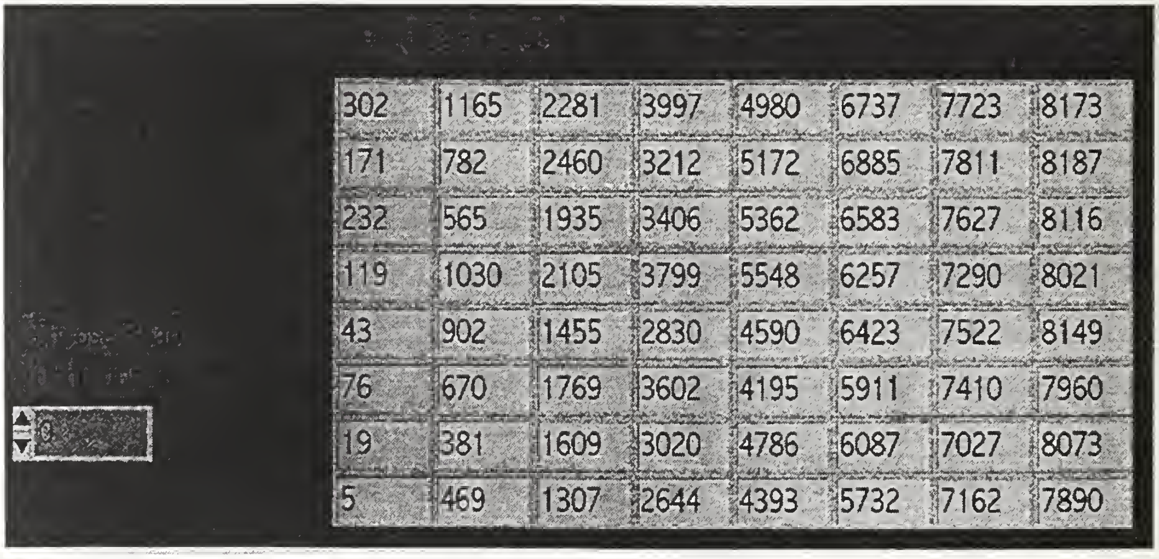


Figure 56   *SNS_Control4g* vi internal diagram (7 of 7).

D. *SNS_RUNF* VI

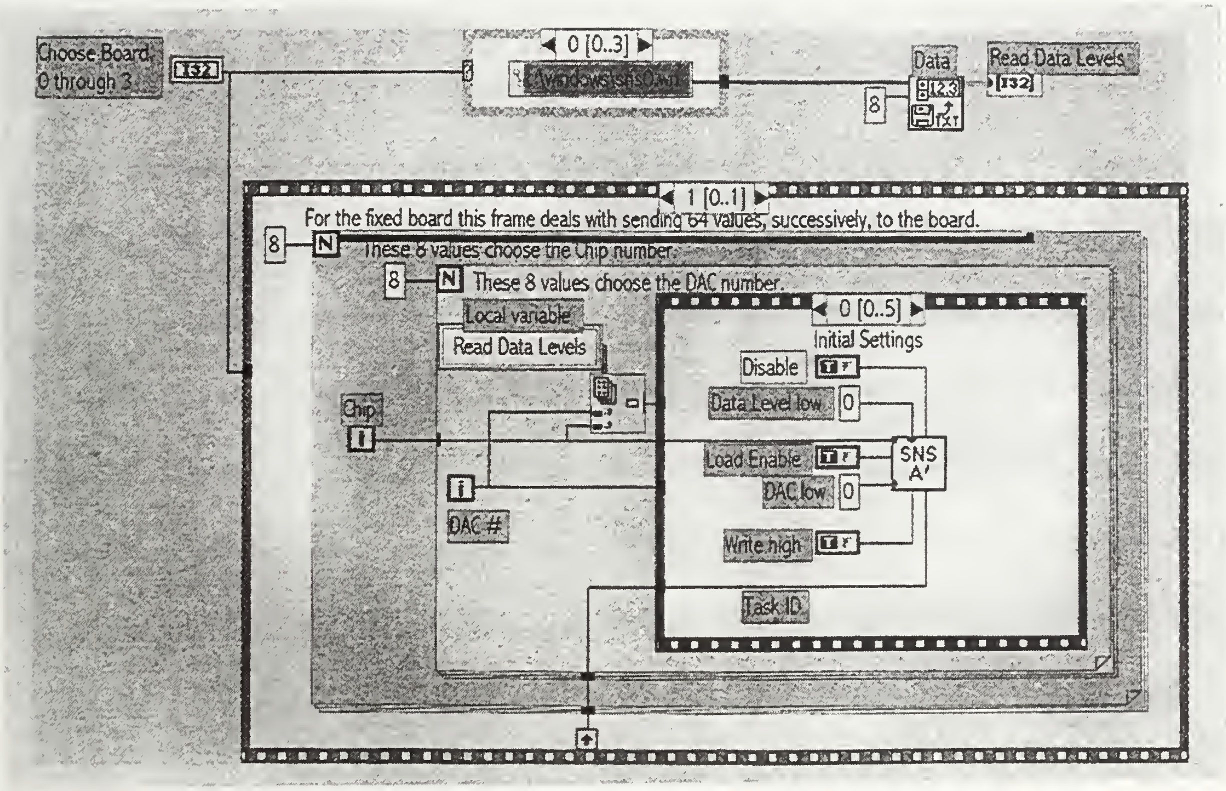
This vi is used to program the comparator boards through the DIO-96 interface board. The front panel (Figure 57) only allows board selection. When the vi is run, it reads the file sns0,1,2,3.wri containing the desired comparator levels, saved by the *SNS_Control4g* vi, and routs the data to the *SNS_Control3f* vi to be sent to the DIO-96.



302	1165	2281	3997	4980	6737	7723	8173
171	782	2460	3212	5172	6885	7811	8187
232	565	1935	3406	5362	6583	7627	8116
119	1030	2105	3799	5548	6257	7290	8021
43	902	1455	2830	4590	6423	7522	8149
76	670	1769	3602	4195	5911	7410	7960
19	381	1609	3020	4786	6087	7027	8073
5	469	1307	2644	4393	5732	7162	7890

Figure 57 ***SNS_Runf*** **vi control panel.**

The internal configuration diagrams for the *SNS_Runf* vi are shown in Figure 58 through Figure 62. Located in the top center of Figure 58 is a Multiframe Numeric Case Structure (MNCS) which works similarly to the BCS described above. This structure is controlled by the Choose Board input and selects the correct file to be read by the LabVIEW *File_Read* vi. The outputs of the *File_Read* vi are displayed on the front panel. The options for the MNCS are shown in Figure 59.



For the fixed board this frame deals with sending 64 values, successively, to the board.

These 8 values choose the Chip number.

These 8 values choose the DAC number.

Figure 58 *SNS_Runf* internal diagram (1 of 6).

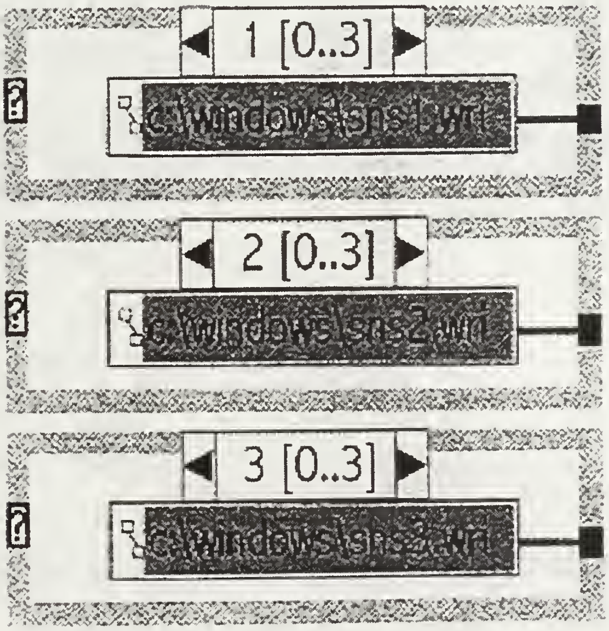


Figure 59 *SNS_Runf* internal diagram (2 of 6).

The large MSS (outer MSS0 is used to either configure the DIO-96 (frame 0 shown in Figure 59) or program the selected comparator board (frame 1 shown in Figure 58.

During the configuration process, the selected board number is multiplied by 3 for addressing considerations, converted to a bit stream and sent to the LabVIEW *Port_Config* vi which generates a Task ID number and routes it back to the outer MSS.

Inside the outer MSS is located two FOR loop structures. These are denoted by the N in the upper right corner of the structure. The structures are used to loop through the chips and DACs on the selected board during the programming process.

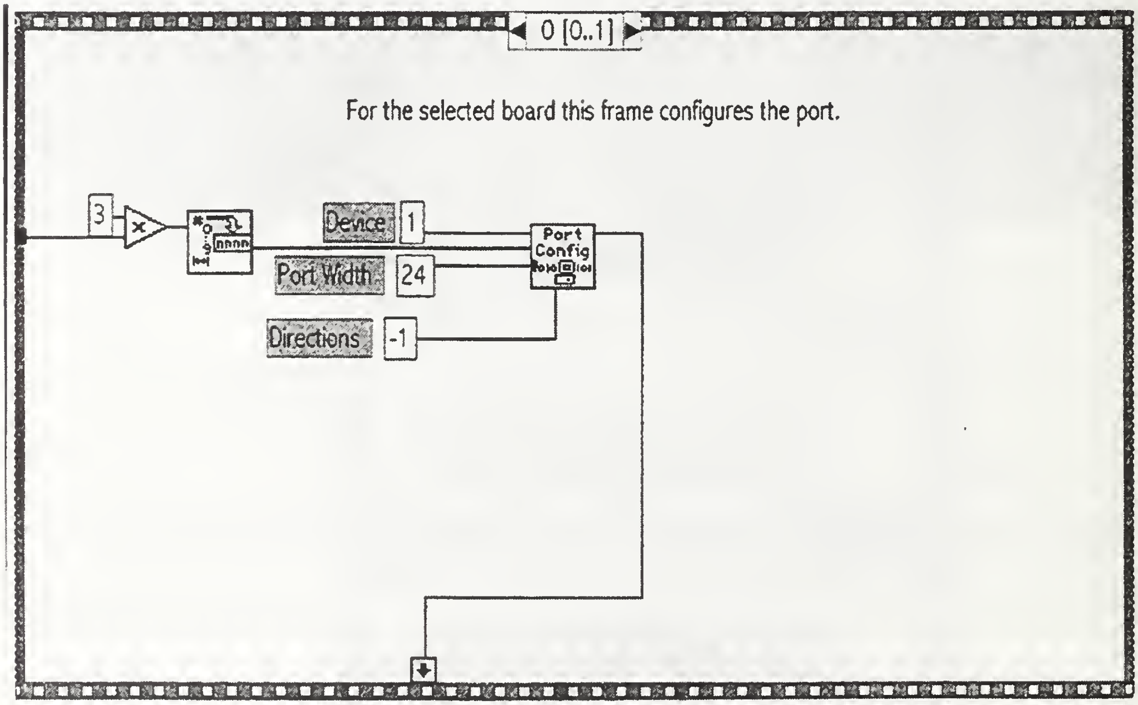


Figure 60 ***SNS_Runf*** **internal diagram (3 of 6).**

The inner MSS performs the operations necessary to program the individual DACs on the MAX547. These steps are shown in Figure 58 (frame 0) and Figures 61 though 65. This programming follows the discussion in Chapter III. Frame 0 (Figure 58) sets the initial conditions for the programming. At this time the chip is already selected. During frame 1 (Fig. 61) the board is enabled, the DAC number is selected and the data value is sent to the board. At frame 2 the Write line is driven low (0) to write the data values to the DAC input latch (Figure 7). Frame 3 (Fig. 62) drives the write line high (1) to isolate the input latches from the data bus and frame 4 (Fig. 62) drives the load enable line low (0) latching the data from the input latches to the DAC latches (Fig. 7). Frame 5 (Fig. 63) resets the initial conditions for the board. After the inner MSS runs through all five frames, the inner FOR loop increments the DAC number and the inner MSS is run again. After eight loops of the inner FOR loop, the outer FOR loop will increment the chip number and the sequence will repeat until all chips and DACs are written to.

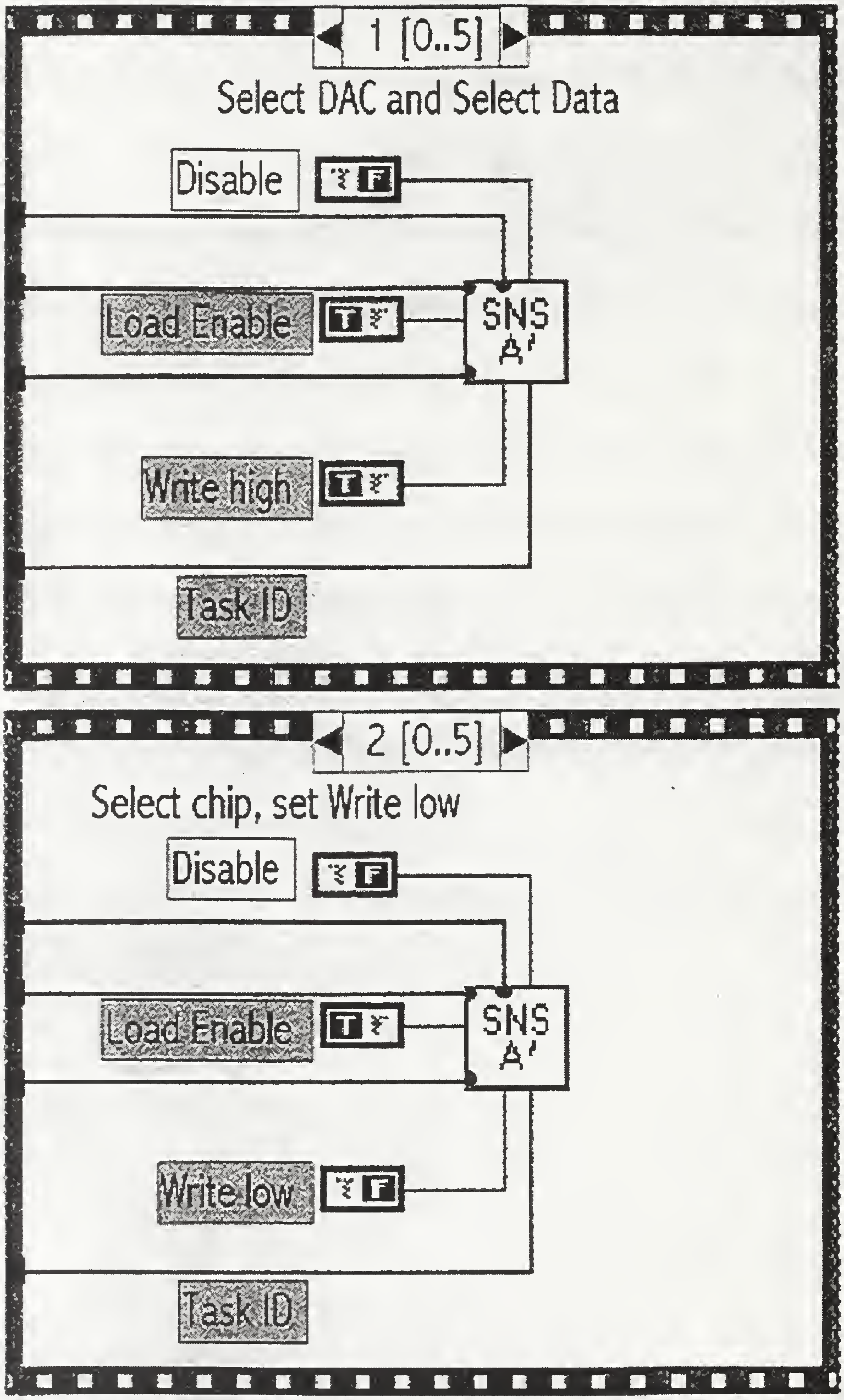


Figure 61 *SNS_Runf* internal diagram (4 of 6).

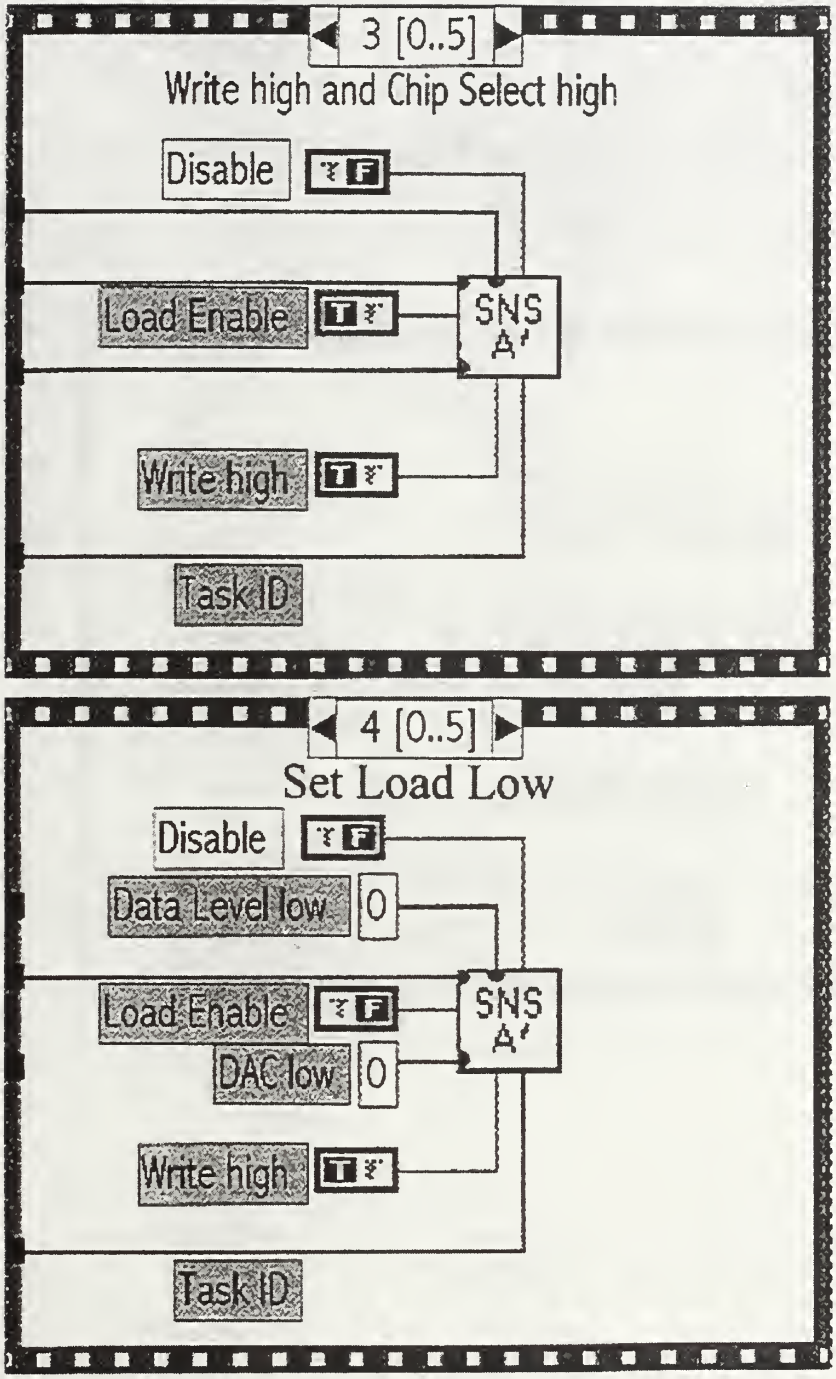


Figure 62 *SNS_Runf* internal diagram (5 of 6).

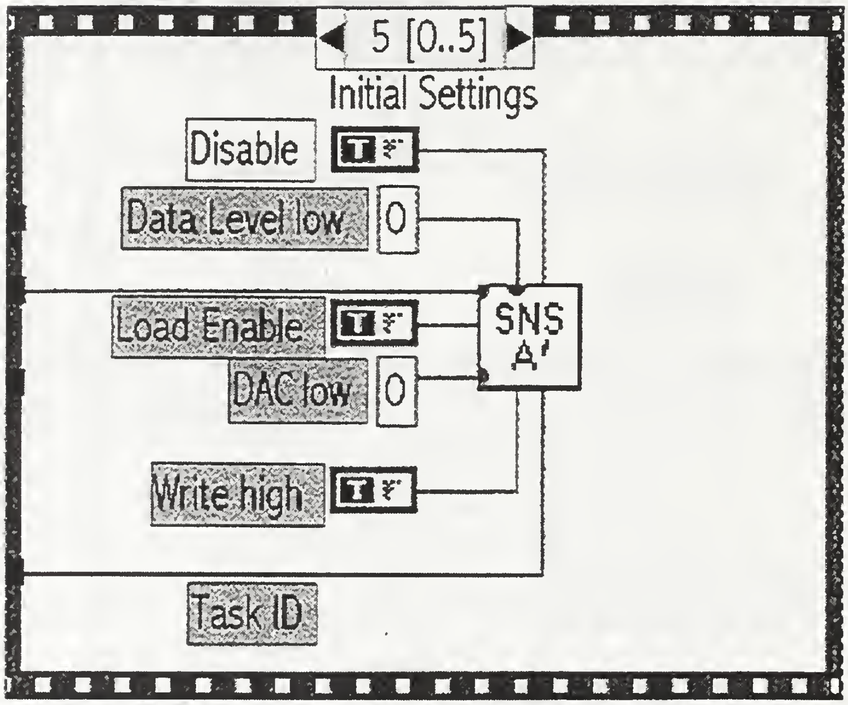


Figure 63 ***SNS_Runf*** **internal diagram (6 of 6).**

APPENDIX D. OSNS FPGA SCHEMATICS

This appendix contains the schematic designs used to implement the OSNS-to-decimal conversion algorithm discussed in Chapter IV. Figure 64 shows the overall schematic with the rest of the figures in the appendix subordinate to this figure.

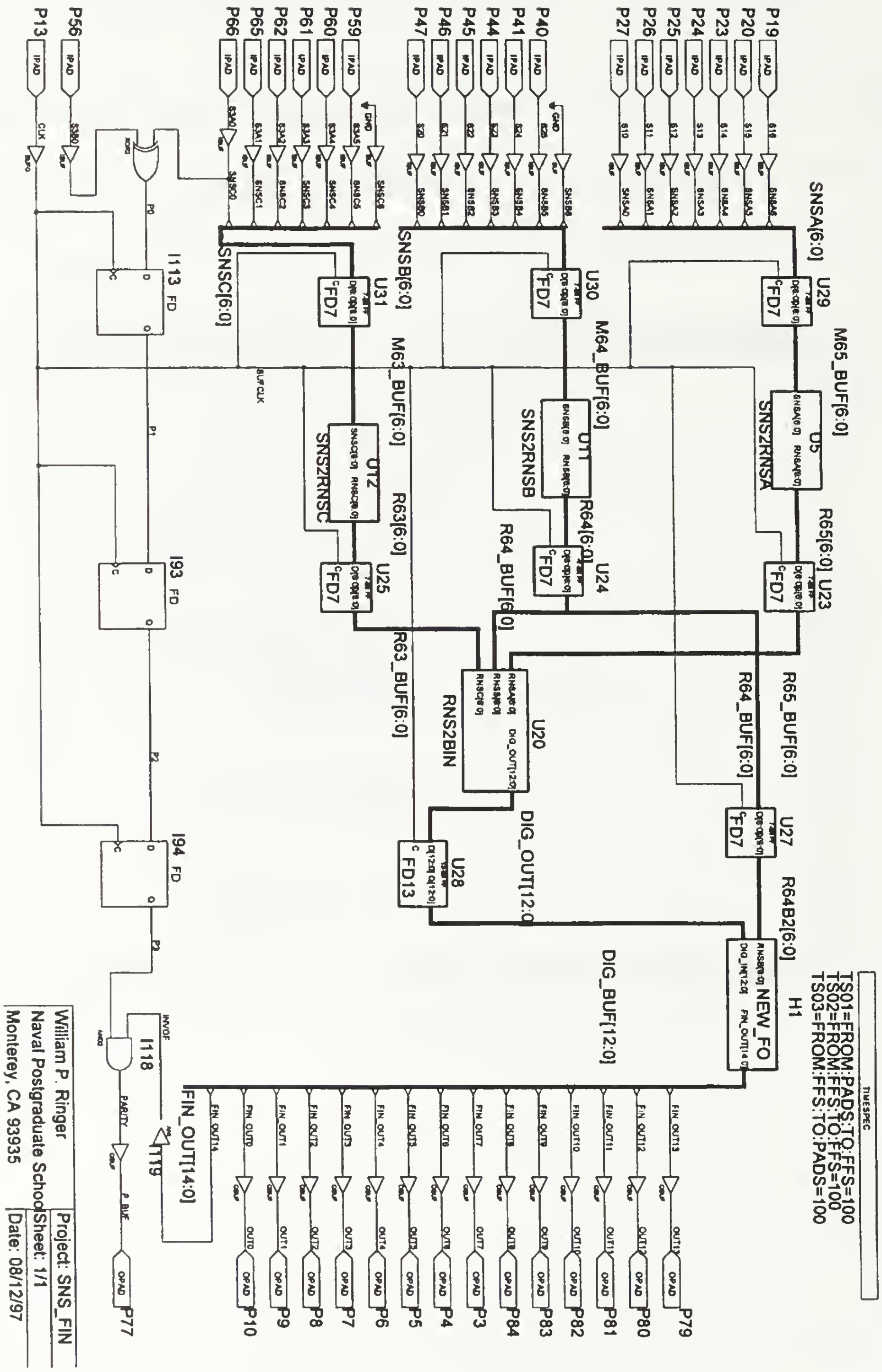


Figure 64 OSNS FPGA design schematic.

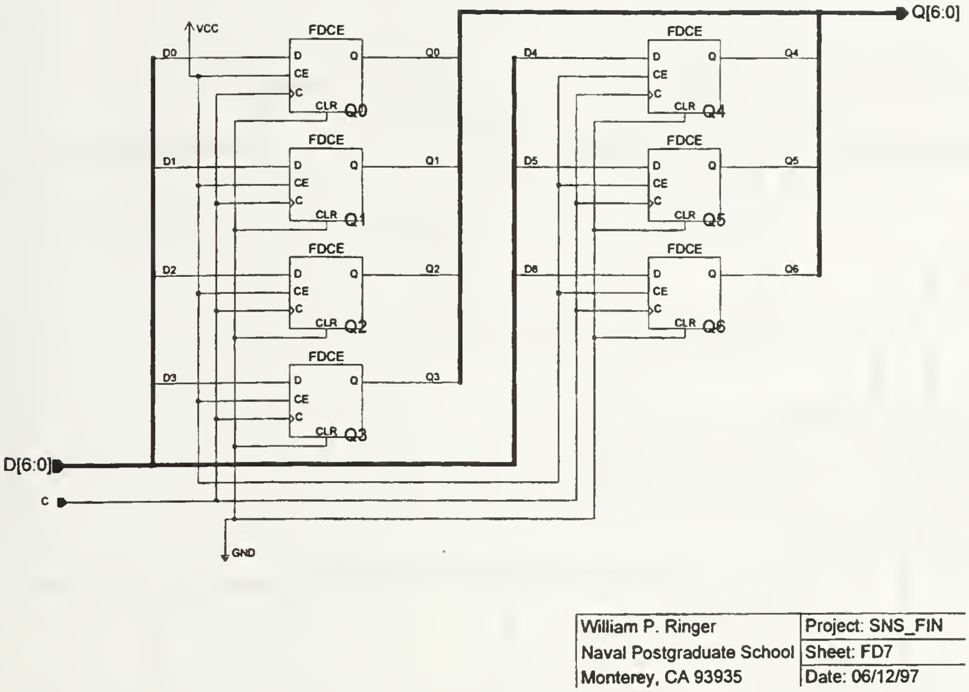


Figure 65 FD-7 7-bit flip-flop latch schematic.

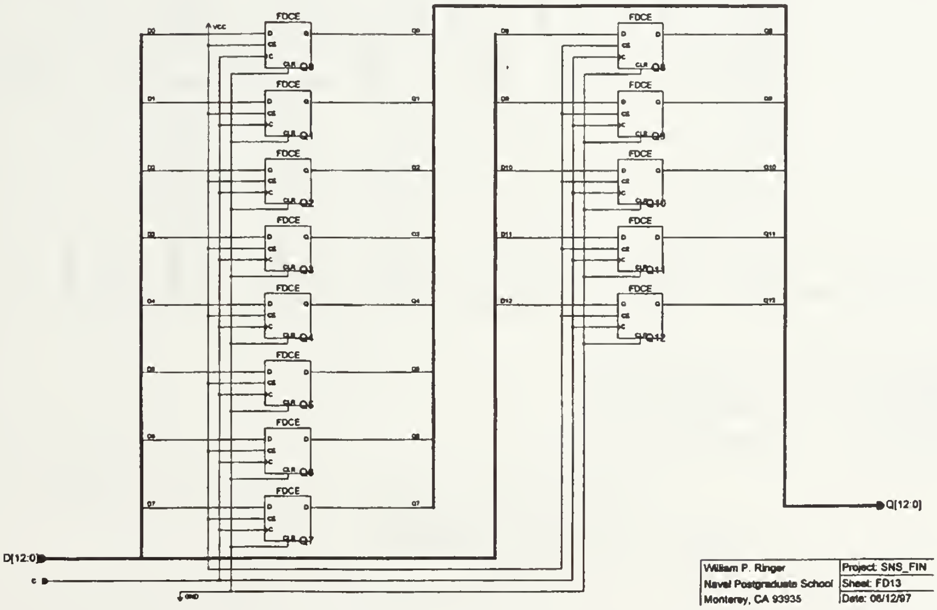


Figure 66 FD-13 13-bit flip-flop latch schematic.

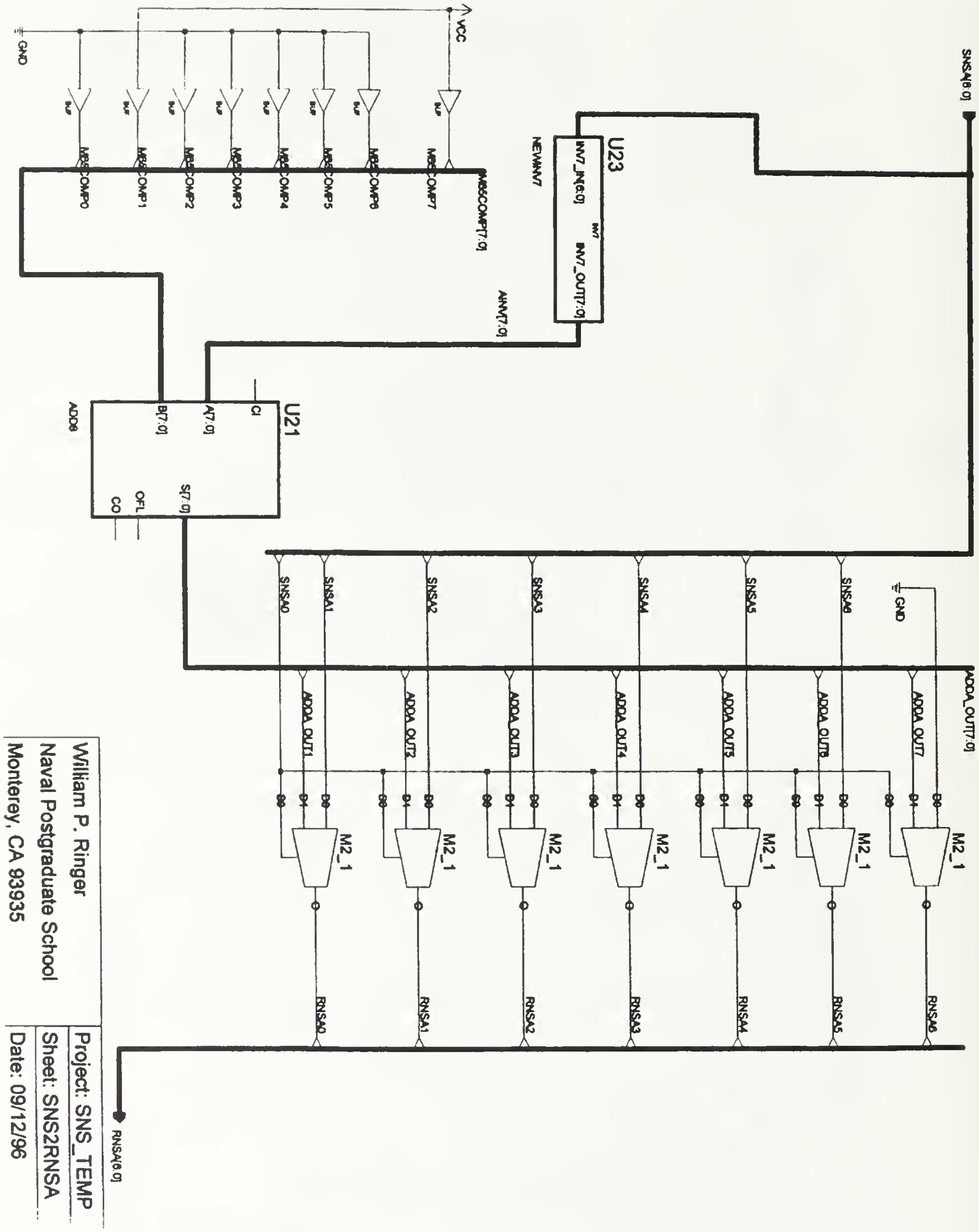


Figure 67 SNS2RNSA schematic.

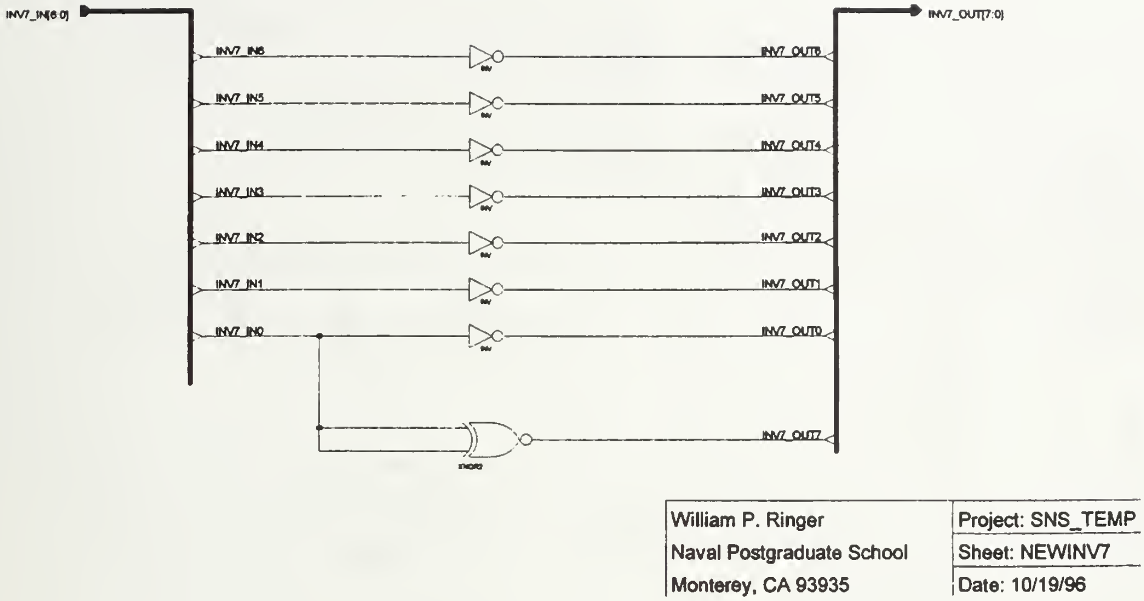


Figure 68 NEW_INV7 schematic.

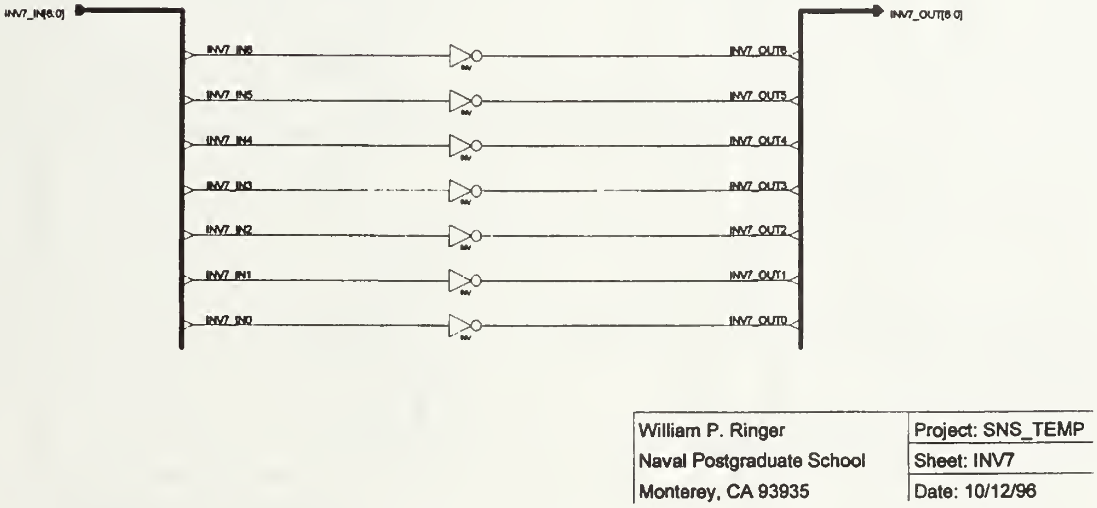


Figure 69 INV7 schematic.

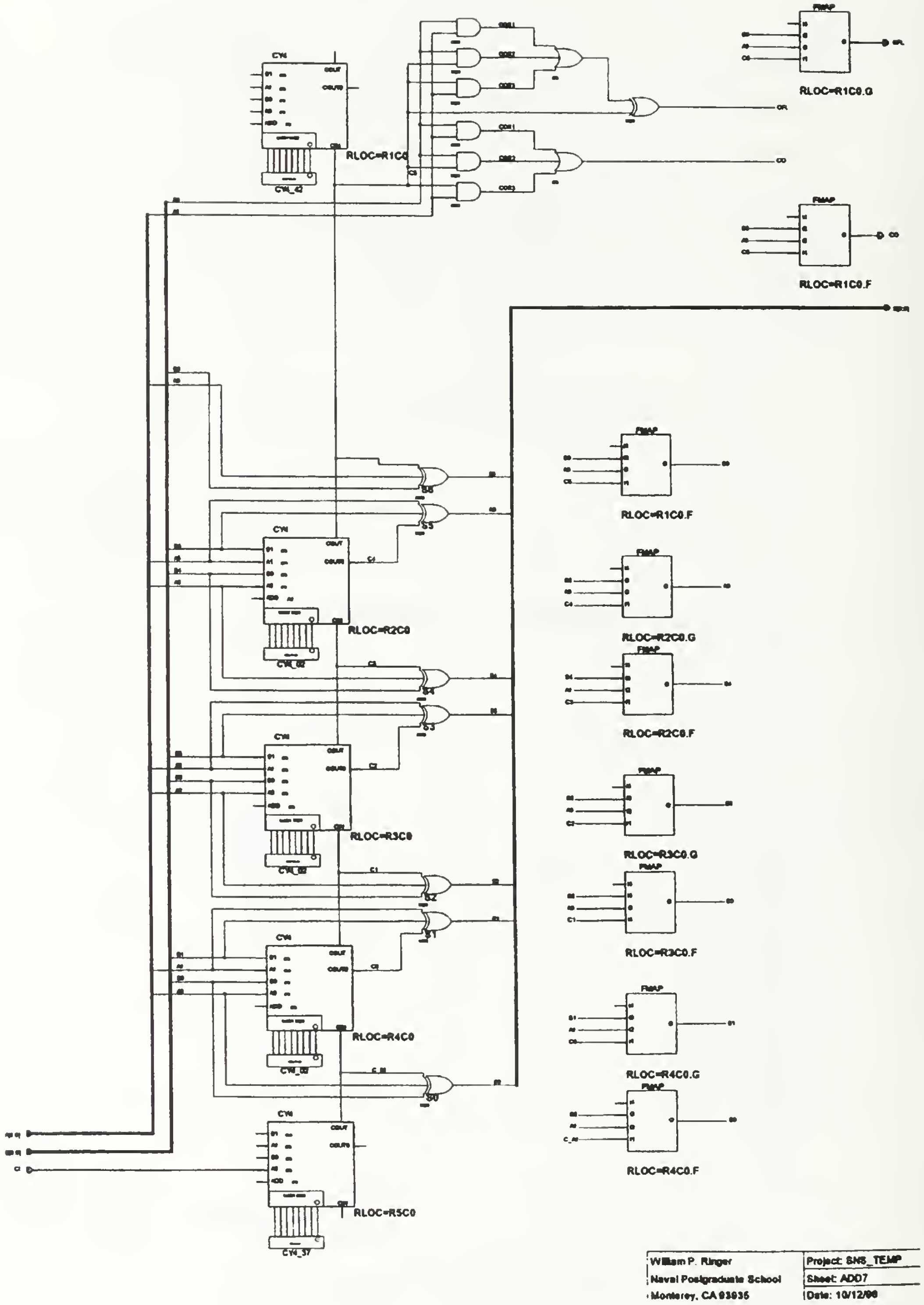


Figure 70 ADD7 schematic

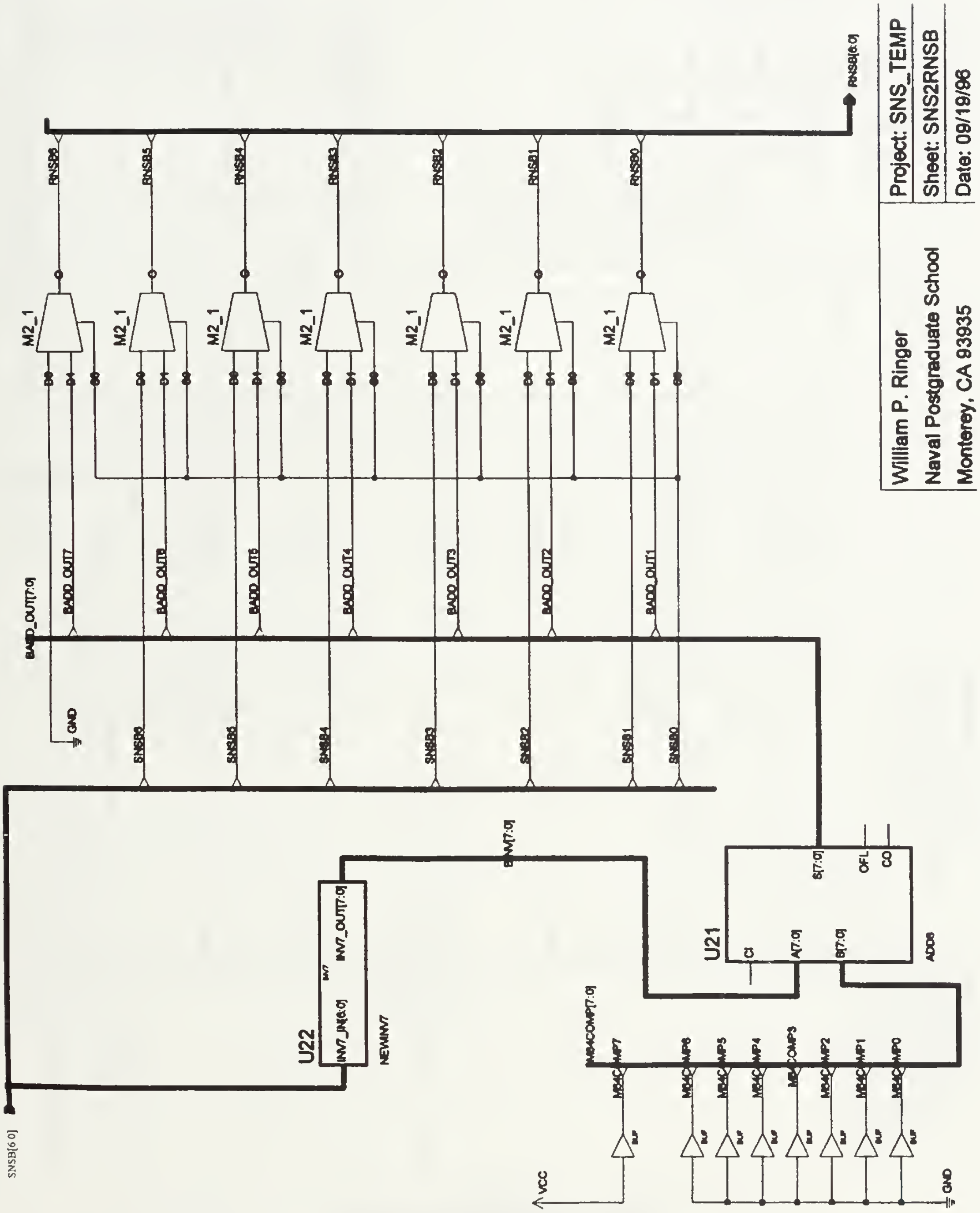


Figure 71 SNS2RNSB schematic.

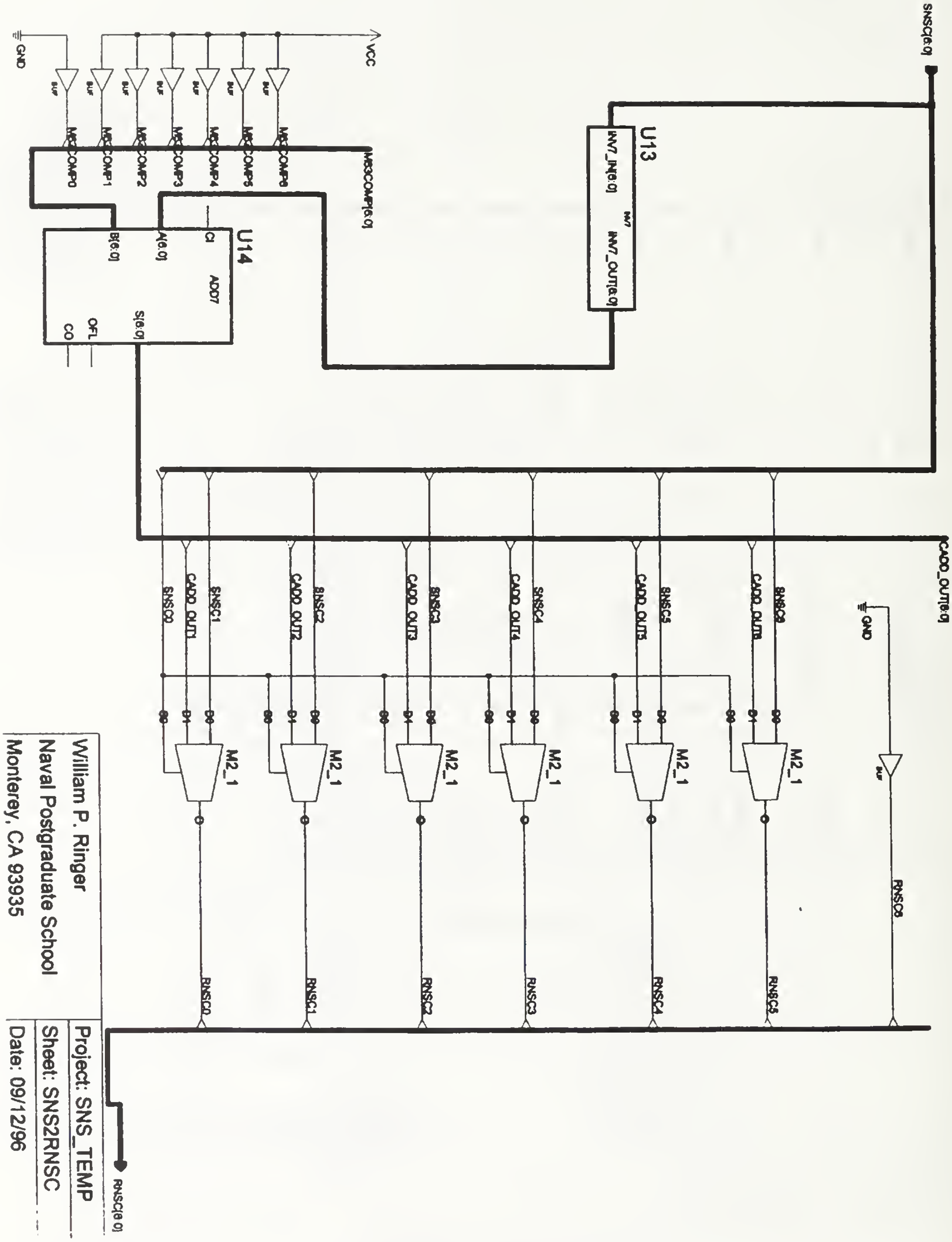


Figure 72 SNS2RNSC schematic.

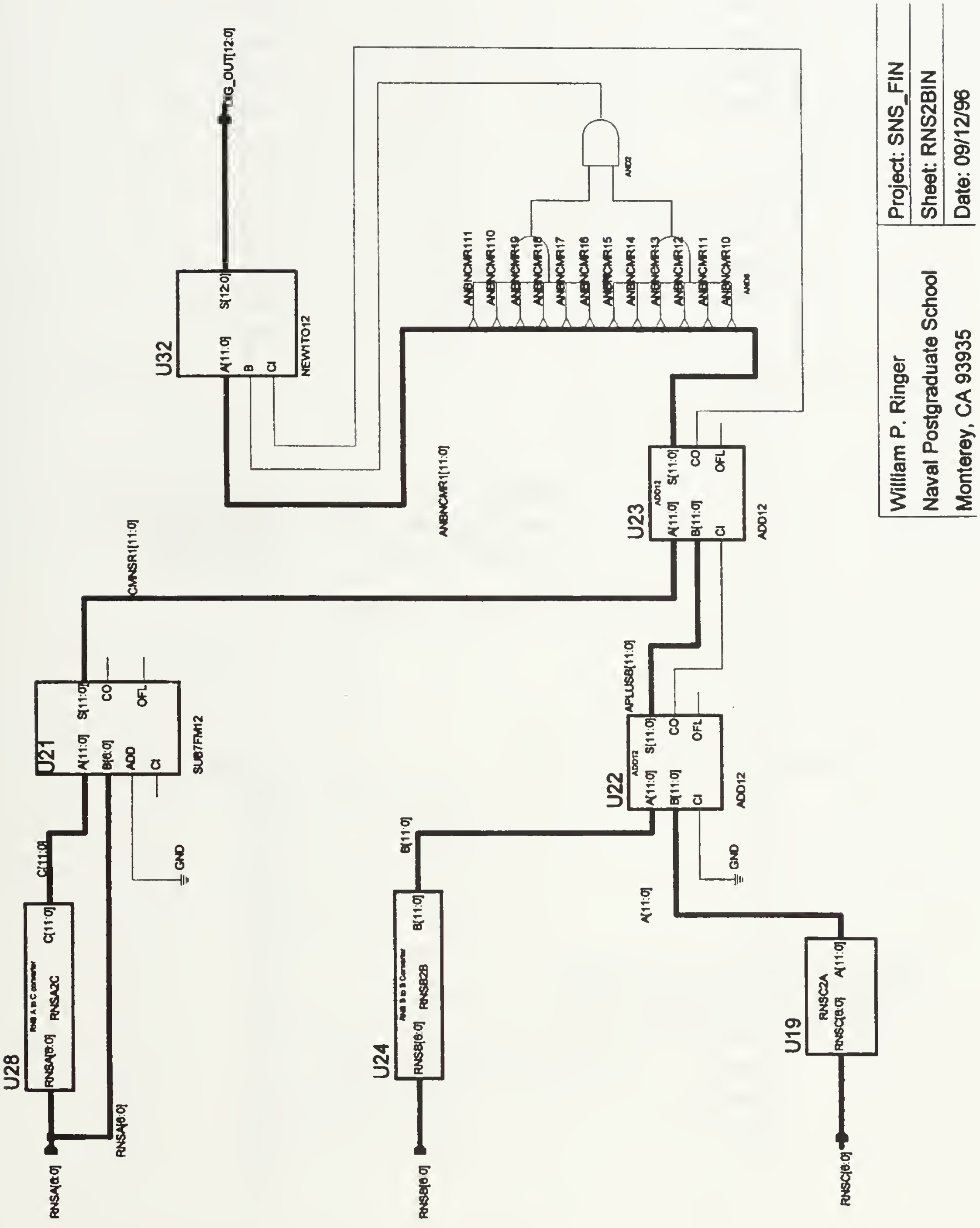


Figure 73 RNS2BIN schematic.

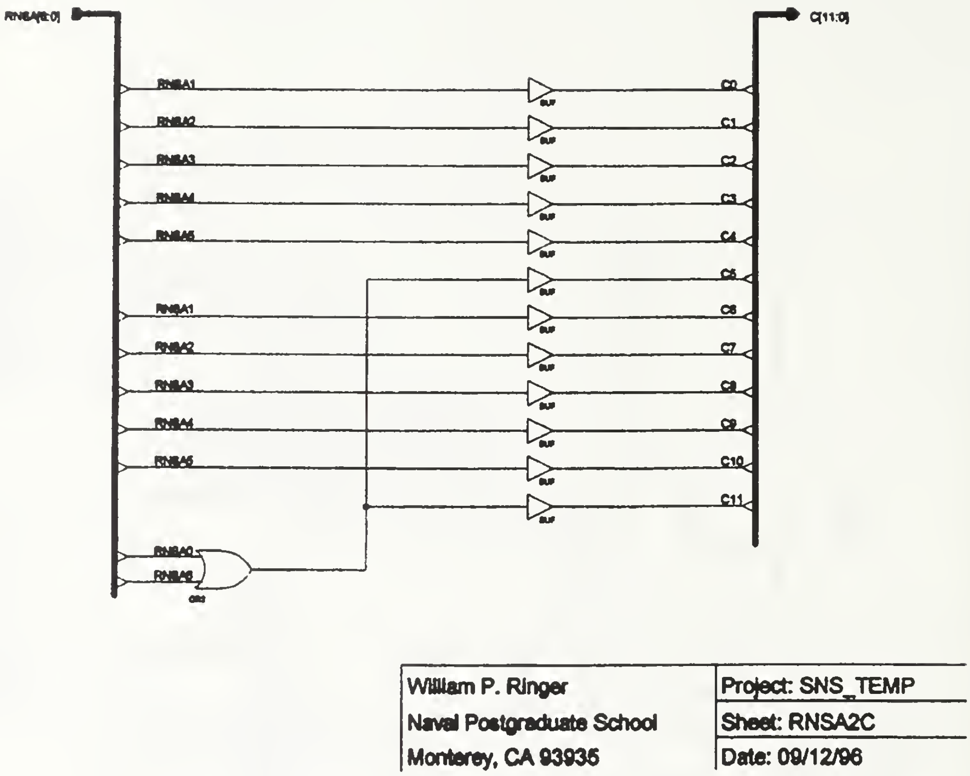


Figure 74 RNSA2C schematic.

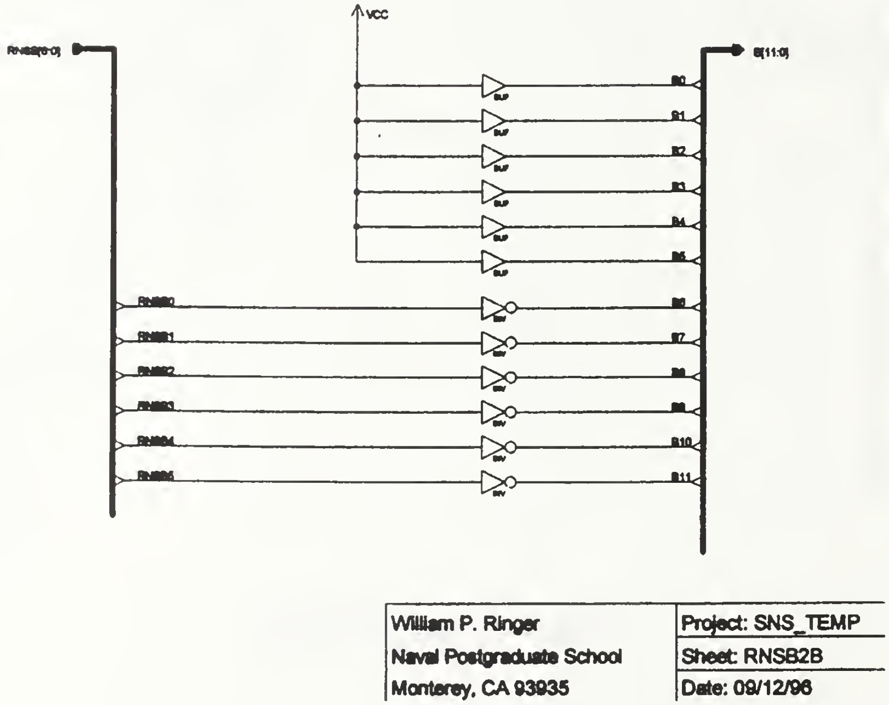


Figure 75 RNSB2B schematic.

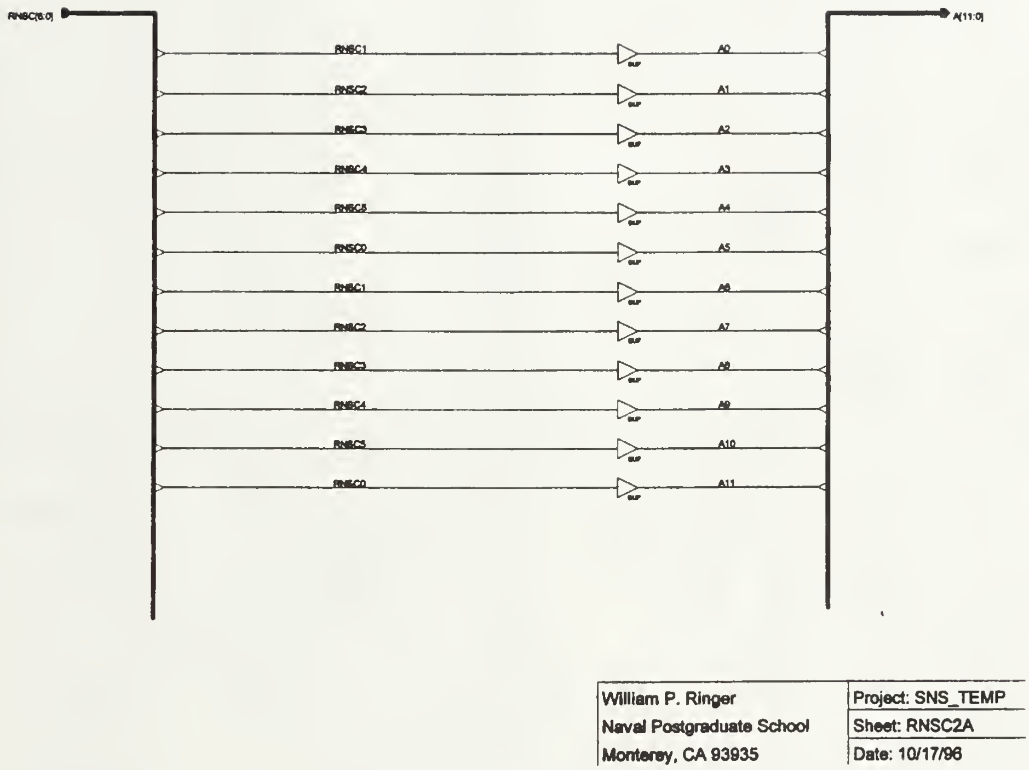


Figure 76 RNSC2A schematic.

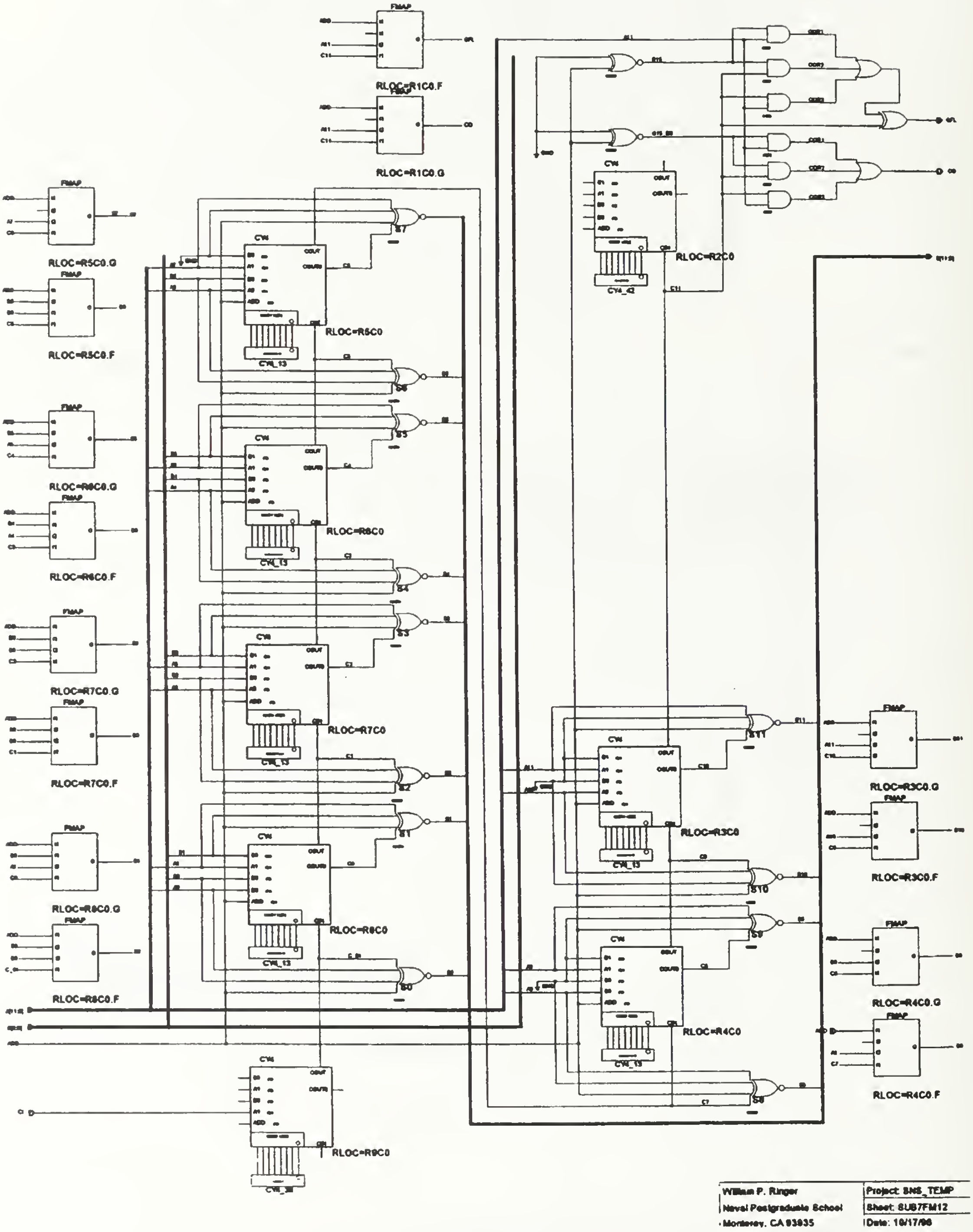


Figure 77 SUB7FM12 schematic.

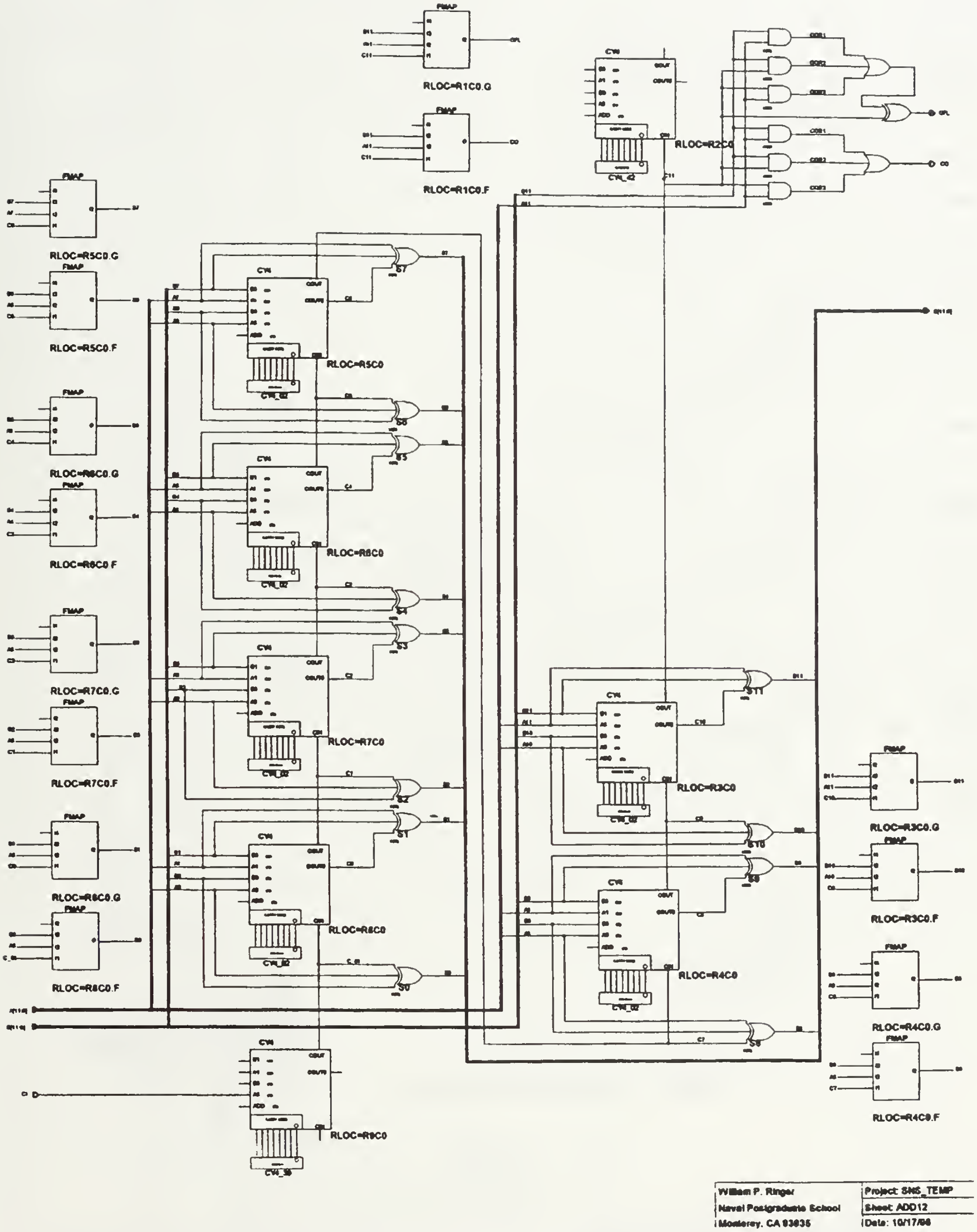


Figure 78 ADD12 schematic.

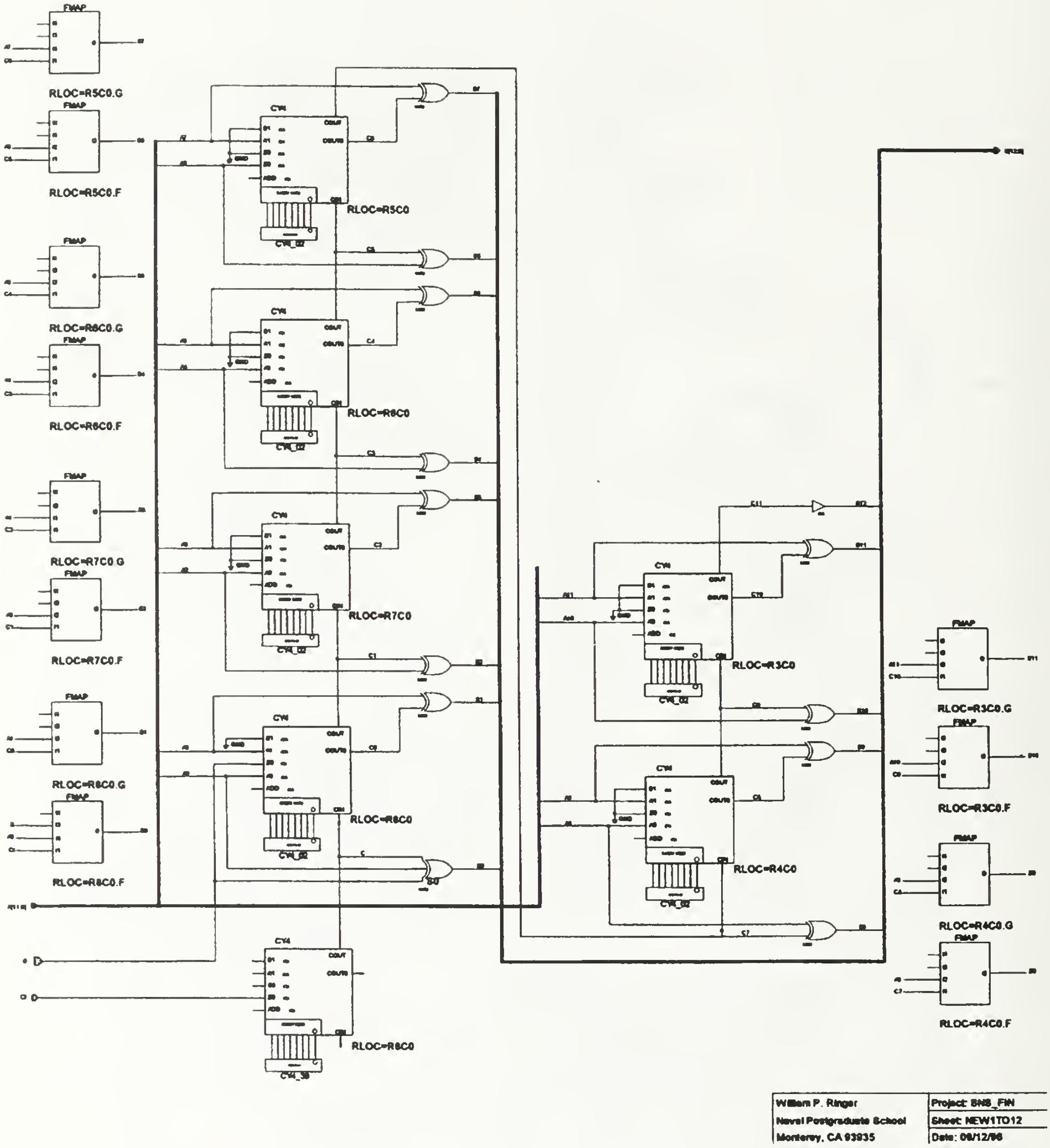


Figure 79 NEW1TO12 schematic.

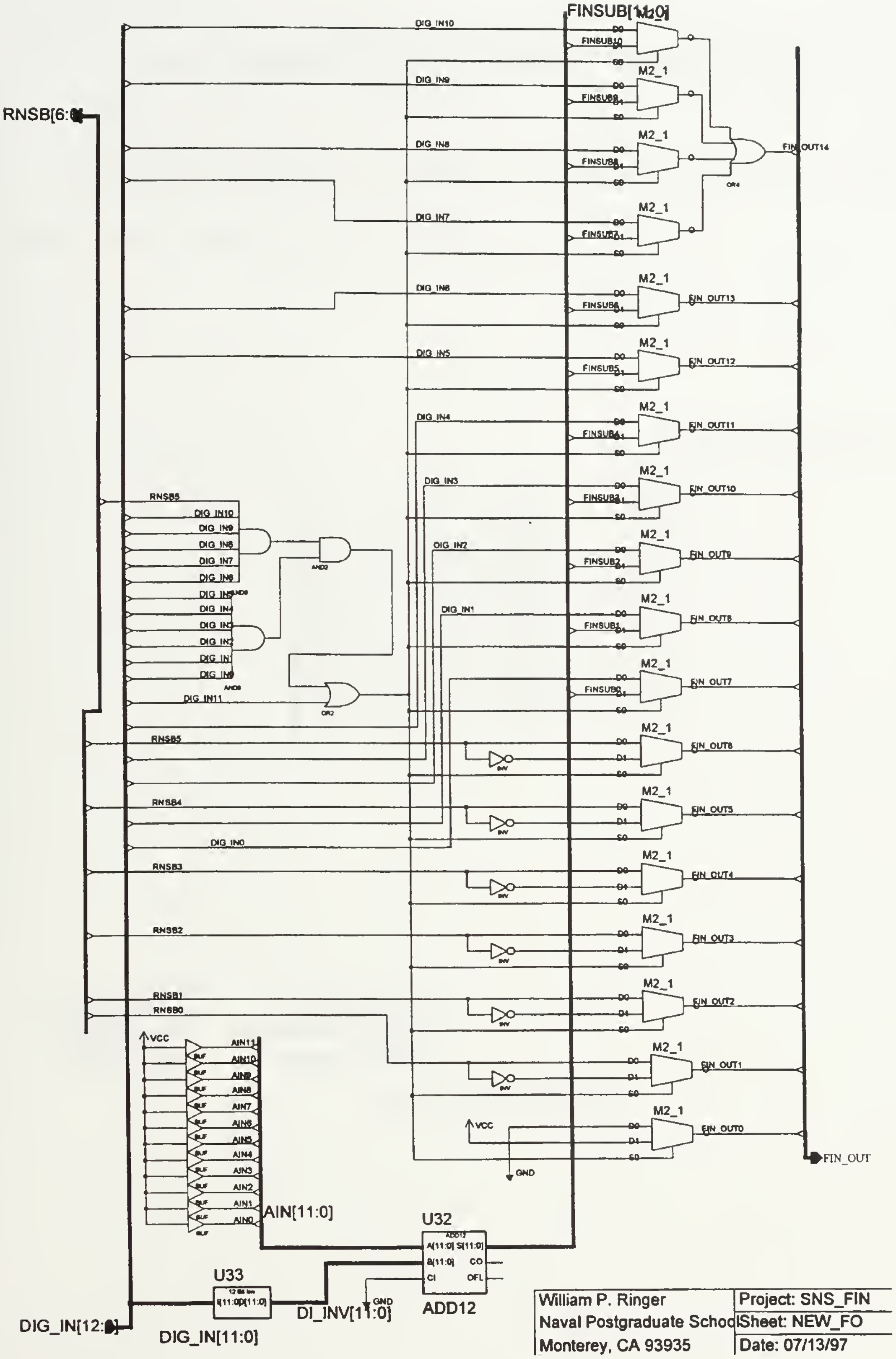
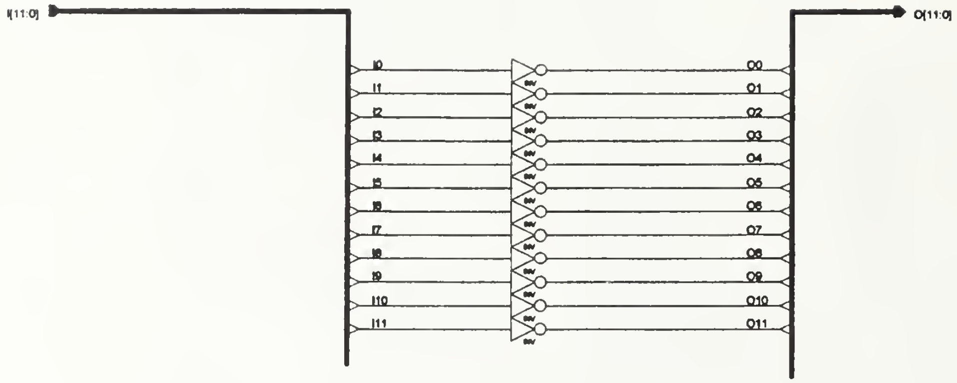


Figure 80 NEWFO schematic.



William P. Ringer	Project: SNS_FIN
Naval Postgraduate School	Sheet: newinv12
Monterey, CA 93935	Date: 09/12/96

Figure 81 INV12 schematic.

APPENDIX E. PARITY FPGA SCHEMATICS

This appendix contains the schematics used to design the parity FPGA. The details of these schematics is given in Chapter 4.

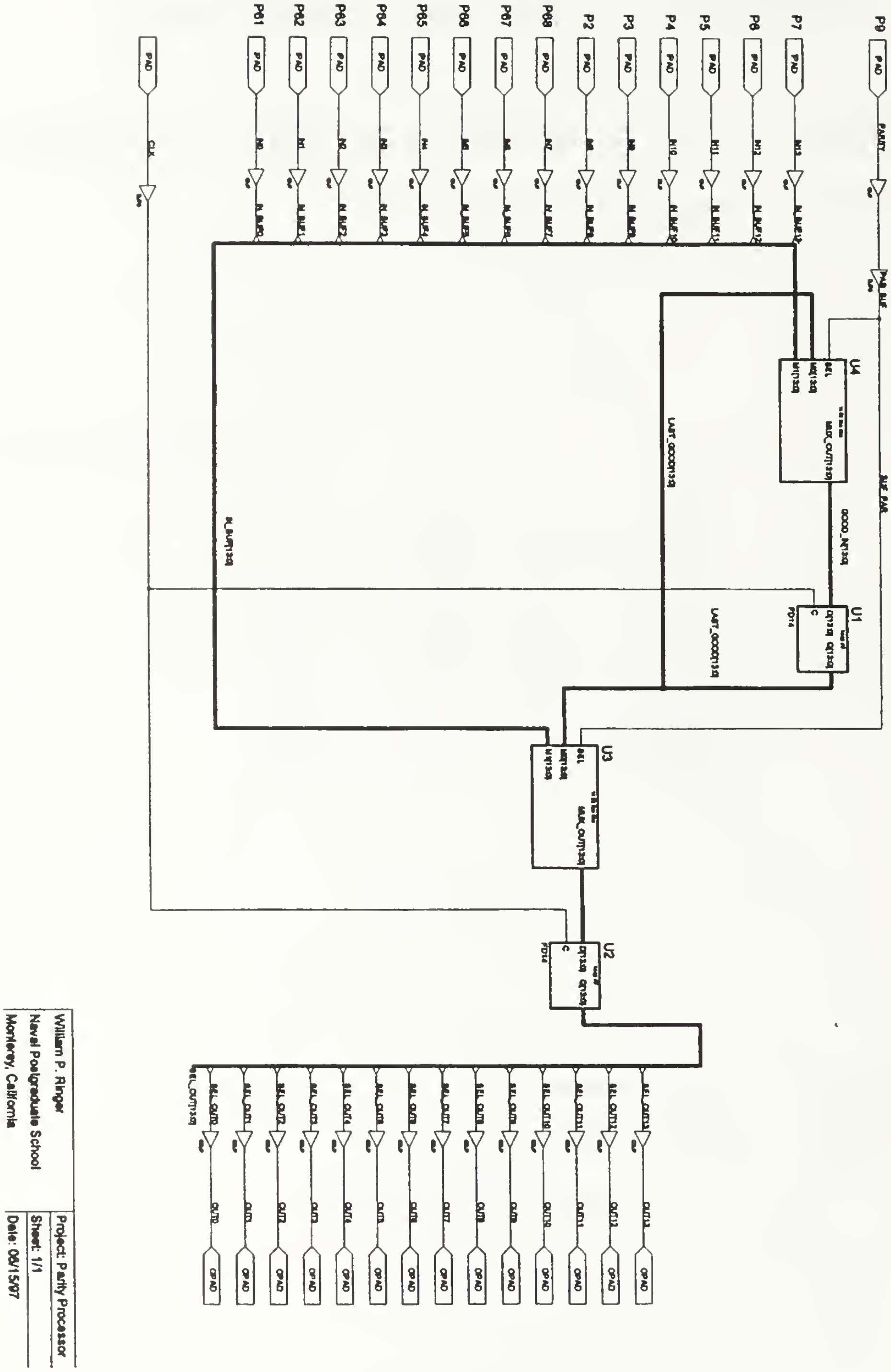
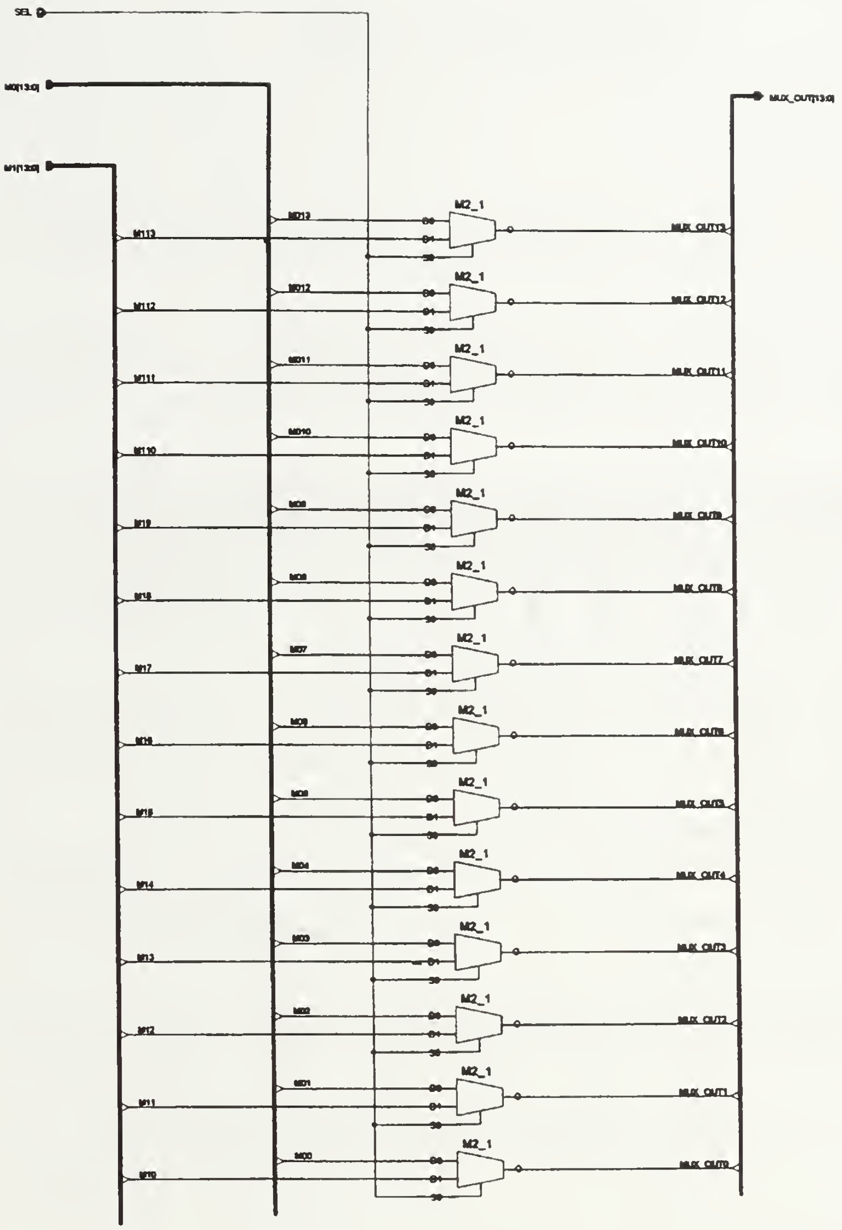


Figure 82 Parity FPGA design schematic.



William P. Ringer	Project: PARITY
Naval Postgraduate School	Sheet: MUX2_14
Monterey, California	Date: 06/16/97

Figure 83 MUX_2_14 schematic.

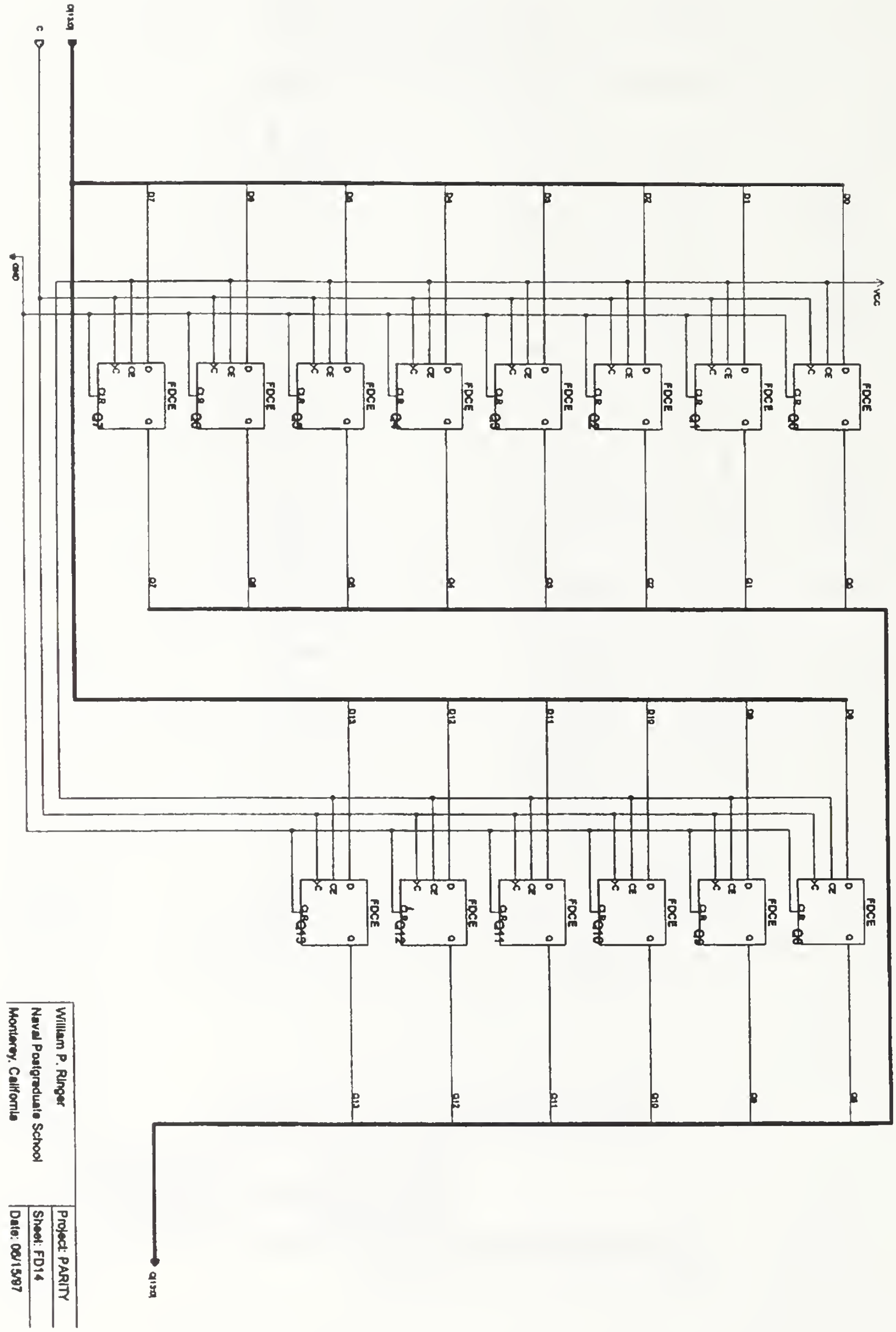


Figure 84 FD14 schematic.

LIST OF REFERENCES

1. Coughlin, R. F. and Driscoll, F. S., "*Operational Amplifiers & Linear Integrated Circuits,*" Prentice Hall Inc., Englewood Cliffs, NJ, 1991.

2. Walley, R. D., "A Prototype Encoding Scheme For Electrooptic Analog to Digital Converters," Master's Thesis, Naval Postgraduate School, Monterey, CA, June 1995.

3. Foster K. D., "Design, Construction, and Analysis of a Optical Preprocessor for a 14 -bit Direct Digital Antenna Using Optical Sampling and Optimum SNS Encoding," Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1997.

4. Pietruszewski, A. P., "A Programmable Offset Differential ECL Four Channel Clock Generator for Applications in High Resolution Digital Antennas," Master's Thesis, Naval Postgraduate School, Monterey, CA, September 1996.

5. Pace, P. E., Styer, D., and Ringer W. P., "Optimum SNS to Binary Conversion Algorithm.", *IEEE Transactions on Circuits and Systems- II: Analog and Digital Signal Processing*, In review.

6. Patterson R.H., "Input Termination Network for a High Resolution Integrated Optical Analog to Digital Converter," NRaD Engineering report, San Diego, CA, October 1995.

7. XILINX, *Hardware and Peripherals Users Guide,* XILINX, San Jose, CA, October 1995.

8. Schokley, J. E., *Introduction to Number Theory*, New York: Holt, Rinehart and Winston, 1967.

9. MAXIM, *Octal, 13-Bit Voltage-Output DAC With Parallel Interface*, Maxim Integrated Products, Sunnyvale, CA, December 1995.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center 2
8725 John J. Kingman Rd., STE 0944
Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library 2
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943

3. Chairman, Code EC 1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943

4. Prof. Phillip E. Pace, Code EC/Pc 3
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943

5. Prof. John P. Powers, Code EC/Po..1
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943

6. Richard Patterson, Code 772 1
NCCOSC-NRaD division
San Diego, CA 92152

7. Lt. Kevin D. Foster, USN 1
944 South Peninsula Drive
Unit 402, Pier 43
Daytona Beach, FL 32118

8. Lt. William P. Ringer, USN..2
15820 Phillips Oak Drive
Spencerville, MD 20868